An Integrated Fuzzy Logic System in A Partially Known Environment

Kyuyeol Shin

Electrical Engineering Program

Graduate School of UNIST

An Integrated Fuzzy Logic System in A Partially Known Environment

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Kyuyeol Shin

01. 24. 2013 of submission
Approved by.

Z. Bien

Major Advisor

Z. Zenn Bien

An Integrated Fuzzy Logic System in A Partially Known Environment

Kyuyeol Shin

This certifies that the thesis of Kyuyeol Shin is approved.

01. 24. 2013 of submission

Signature

Thesis Supervisor: Z. Zenn Bien

Signature

Thesis Committee Member #1: Gil-Jin Jang

Signature

Thesis Committee Member #2: Jae-Young Sim

Abstract

In this thesis, we deal with a learning method for mimicking human behaviors in a partially known environment. Humans are possible to infer the appropriate action based on the partially known information to achieve their goals. The fuzzy logic is able to deal with a process of human reasoning by describing IF-THEN rules. The FQL method which is one of the reinforcement learning methods uses fuzzy logic for decision making. FQL is possible to expect improvement of the learning efficiency rather than ordinary Q-Learning method by using fuzzy logic. However, the problem comes from a conclusion part of FQL. The conclusion part of FQL consists of a set of singleton values. The behaviors of the agent are influenced by the number of singleton values. Especially, it is hart to expect the efficiency of learning when the environment is complex and a precise behavior is required. So, we suggest an integrated fuzzy logic system which is FQL with ANFIS.

Contents

I	**INTRODUCTION**	**1**
II	**PROBLEM DESCRIPTIONS ANS SURVEYS**	**5**
	2.1 Problem Descriptions	5
	2.1.1 Reasoning of Human Behaviors	5
	2.1.2 Problem Descriptions	7
	2.2 Surveys	11
	2.2.1 Fuzzy Inference System	11
	2.2.2 Q-Learning	18
	2.2.3 Fuzzy Q-Learning	20
	2.2.4 Adaptive Network based Fuzzy Inference System	24
III	**AN INTEGRATED FUZZY LOGIC SYSTEM**	**30**
	3.1 Descriptions	30
	3.2 FQL with ANFIS	33
IV	**SIMULATION**	**40**
V	**CONCLUSION**	**50**
ACKNOWLEDGEMENTS		**53**

List of Figures

Figure 1.1	(a) Intelligent Sweet Home (b) Steward Robot Joy	1
Figure 1.2	An inverted pendulum system designed by fuzzy logic	3
Figure 2.1.1	Human reasoning in the game of darts	5
Figure 2.2.1	The structure of a Fuzzy Rule-based System	11
Figure 2.2.2	The strong fuzzy partition of a fuzzy variable for *distance*	12
Figure 2.2.3	Membership function of linguistic words in Example 4	14
Figure 2.2.4	A Mamdani type FIS with two inputs and a single output	15
Figure 2.2.5	A TSK type FIS with two inputs and a single output	16
Figure 2.2.6	The system architecture of ANFIS	23
Figure 3.1.1	The mountain car problem	29
Figure 3.1.2	The simple navigation problem	29
Figure 3.1.3	The avoiding actions of the agent	30
Figure 3.2.1	The isosceles triangular membership functions	31
Figure 3.2.2	The local singleton q-values in the conclusion part	32
Figure 3.2.3	The EEP algorithm for selecting the local action	32
Figure 3.2.4	The trapezoidal membership shape for reinforcement signal	33
Figure 3.2.5	The Bell membership function shape	33
Figure 3.2.6	The change of membership function shape caused by parameters.	34
Figure 3.2.7	Updating premise parameters by gradient descent method	35
Figure 3.2.8	The system flow chart of FQL with ANFIS	37
Figure 4.1	The advance information of the putting green in real world	38

Figure 4.2	The modeled putting green and GUI of FQL with ANFIS	39
Figure 4.3	The structures of ANFIS regarding the actual outputs of velocity and direction	42
Figure 4.4	The fuzzy variables of the putting strength and direction	43
Figure 4.5	The best three cases of ordinary Q-Learning	44
Figure 4.6	(a) The average number of the putting when the number of decisions is four (b) The histogram of the first episode	45
Figure 4.7	The trajectory of the ball when making a hole in one	46
Figure 4.8	The number of the putting when the decision is 1 x 1	47
Figure 4.9	(a) The first situation (b) The second situation	47
Figure 4.10	The human behavior learning based on using the learned data in the similar environment.	48

List of Tables

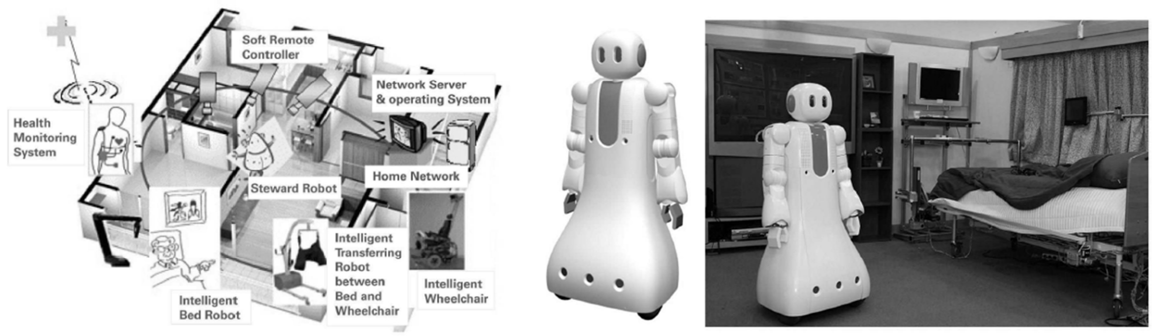
Table 1	The hybrid learning procedure for ANFIS	25
Table 2	The fuzzy rules of FQL	42
Table 3	The membership degrees of observed states	43
Table 4	The best three cases of ordinary Q-Learning	44
Table 5	The simulation result of 2 x 2 decisions FQL with 3 linguistic words of ANFIS	46
Table 6	The simulation result of 2 x 2 decisions FQL with 5 linguistic words of ANFIS	46

List of Abbreviations

AIBFC	Agglomerative Iterative Bayesian Fuzzy Clustering
ANFIS	Adaptive Network based Fuzzy Inference System
ANN	Artificial Neural Network
CMAC	Cerebellar Model Articulator Controller
ECG	Electrocardiogram
EEP	Exploration/Exploitation Policy
EMG	Electromyogram
FCM	Fuzzy C-Means
FIS	Fuzzy Inference System
FQL	Fuzzy Q-Learning
GMM	Gaussian Mixture Model
ISH	Intelligent Sweet Home
LSE	Least Square Estimator
MIMO	Multiple Inputs-Multiple Outputs
MISO	Multiple Inputs-Single Output
NN	Neural Network
SARSA	State-Action-Reward-State-Action
SOM	Self-Organizing Map
SVM	Support Vector Machine
TD	Temporal Difference
TSK	Takagi-Sugeno-Kang

WHO	World Health Organization

I. INTRODUCTION



(a) **(b)**

Figure 4.1: (a) Intelligent Sweet Home. (b) Steward Robot Joy.

Most of the modern people are spending a busy day and they get a lot of stress in daily life. It means that there is more demand for support in human life to take a break. For instance, the application of smart home technology is an attempt to improve the quality of human life. There are various kinds of smart home such as, Adaptive House, Easy-Living, and House of the future [Mozer, MC 1998; Brumitt, B 2000; House_n]. A lot of sensors which are connected each other through wireless network are installed in smart home for monitoring user's behaviors and the state of health. Moreover, the smart home technology is one of the alternatives to ensure the independent living of the elderly and the disabled. It is reported that more than a billion people in the world experience disability in "World Report on Disability" [WHO 2011]. According to the report, the increasing population of the elderly and the disabled is caused by extending the average span of human life and growing people who is suffered from chronic diseases such as, *diabetes*, *cardiovascular disorder*, *cancer*, and *mental disease*. Most of the elderly and the disabled have restrictions on their movements due to physical constraints so it is hard to live alone. In point of view, intelligent robotic devices or systems can help the elderly and the disabled to achieve the independent living. ISH in Figure 1.1 (a) which is developed at the Korea Advanced Institute of Science and Technology is a typical example to help the independent living of the elderly and the disabled [Bien, ZZ 2008; Lee, J-J 2007].

There are a lot of types of intelligent robots and systems in ISH such as, intelligent bed robot, intelligent wheelchair, steward robot, system for transferring between bed and wheelchair, soft remote control system, home network and health monitoring system. These systems can provide the elderly and the disabled with various convenient services. However, it could be hard to control thoroughly for the elderly and the disabled because of a number of intelligent systems and physical constraints on their body. In this case, an intelligent robot which has learning capabilities is one of alternative. For instance, a steward robot in Figure 1.1 (b), named Joy, which is human-friendly intelligent robot, is developed to help users in ISH [Bien, ZZ 2008; Lee, H-E 2006; Park, K-H 2007]. The steward robot can provide users (the elderly and the disabled) with more convenient services by interacting with them as an intermediate system between users and intelligent devices in ISH. Also the steward robot can understand a situation of users and perform an appropriate action by recognizing patterns of user's behavior and service preferences through learning. Thus, it is very important that an intelligent robot has learning capabilities like human.

A recent study [Lee, SW 2008 in Chapter V] is deal with a learning algorithm in uncertain sequence based on the observed data regarding patterns of human behavior, such as ECG, EMG, life cycle etc. Traditionally, these patterns are very complex and uncertain and ambiguous. In the study, the measured pattern data of human behavior from sensors is classified automatically by the proposed classification method (AIBFC) which is modified version of FCM [Lee, SW 2008 in Chapter IV]. Traditionally, the fuzzy method is well known to deal with ambiguity and uncertainty and to describe human reasoning and knowledge processing as linguistic terms based on IF-THEN rules. For this reason, the fuzzy logic concept is applied to express *a priori knowledge* that is people like.

We give an example for a real system, which is an inverted pendulum, applied fuzzy logic to represent fuzzy method more specifically. An aim of the inverted pendulum system is keeping a stick not to fall down left side or right side. The cart that has sustained the stick ought to move both sides to keep standing the stick. To realize this system, we need to model a relationship between input and output as the mathematical equation following Newton's law. Typically, it is not easy works. On the other hand, the fuzzy logic is possible to handle modeling easily by describing linguistic words based on human reasoning. The realization of the inverted pendulum system designed by fuzzy logic controller is shown in Figure 1.2 [Bien, ZZ 2011]. The fuzzy logic controller consists of several fuzzy rules. Thus, the fuzzy method can be considered as one of nonlinear modeling method.

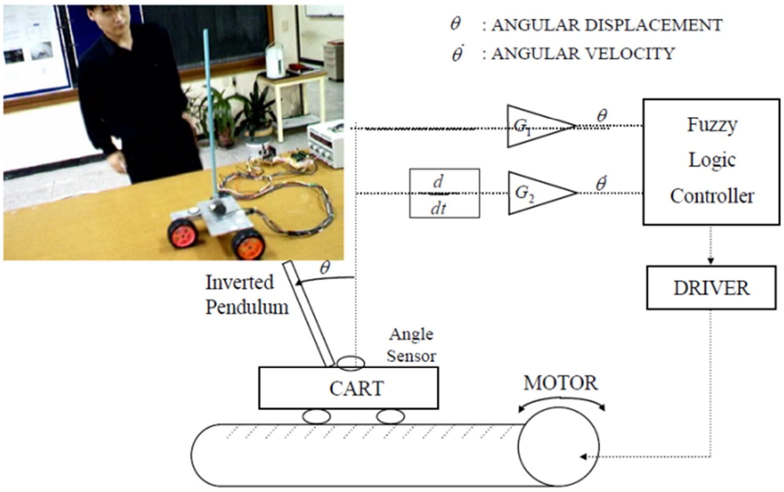


Figure 1.5: An inverted pendulum system designed by fuzzy logic.

Intelligent robot is required to comport themselves well in an uncertain/unknown environment even if optimal behavioral strategies are not pre-planned. Since humans have learning abilities, they can cope with crisis in an uncertain/unknown environment. Machine learning is a branch of artificial intelligence and studies of computer systems for giving learning capabilities to machine like humans. There are several types of learning methods such as, supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

1) *Supervised Learning* is a learning method that can learn the relation between input and output pair form labeled training data. The training data consist of a set of training vectors and a set of desired output values regarding each training vector. It is called a classifier if the output of learned system is discrete or it is called a regression function if the output of learned system is continuous. There are some examples of supervised learning algorithm, such as ANN, SVM, etc.

2) *Unsupervised Learning* is a learning method to find the structure from given unlabeled data. Since the given examples unlabeled, there is no desired output value. This distinguishes unsupervised learning from supervised learning. Examples of unsupervised learning are clustering algorithms, such as SOM, GMM.

3) *Semi-supervised Learning* is a method that combines both labeled and unlabeled data for training. Typically, there are a small number of labeled data in training data. The advantage of semi-supervised learning is that learning accuracy is improved by combining a small number of labeled data and a large amount of unlabeled data.

4) *Reinforcement Learning* is a learning method that an agent (learner) searches an optimal strategy

in an uncertain/unknown environment. The agent takes action when a state vector is given. The state vector consists of a set of data to express a situation of the agent. The uncertain/unknown environment gives the reinforcement signal to the agent. It is a punishment or a reward regarding taken action. Reinforcement learning is that the agent learns which action makes the maximum cumulative reinforcement signal in the given state vector at the uncertain/unknown environment.

Thus, reinforcement learning learns how the agent should take actions in an uncertain/unknown environment in order to maximize cumulative reward through trial and error [Kaelbling, LP 1996]. In point of view, we use partially known information described by fuzzy logic and reinforcement learning method for mimicking human reasoning. The uncertain/unknown environment is partially known as IF-THEN rules of fuzzy logic. IF-THEN rules allow expressing the human knowledge processing and are easy to deal with uncertainty and ambiguity. Moreover, partially known information is represented by means of linguistic terms. It means that the concept of fuzzy logic is very readable and easy to introduce *a priori knowledge* in IF-THEN rules. So, partially known information is applied to *a priori knowledge* when the agent learns an optimal behavior regarding observed states.

In this thesis, we use FQL method for mimicking human behavior. FQL is a learning method which is a combination version of FIS and Q-Learning. FIS is designed by expressing an appropriate behavior that humans make a decision in an uncertain/unknown environment. Next, Q-Learning evaluate the predict behavior by FIS and learn which behavior is made for the maximum cumulative reinforcement signal in a partially known environment. Furthermore, we propose an integrated fuzzy logic system which is combination version of FQL and ANFIS to learn human reasoning more effectively in given states. ANFIS is a learning method that is a combination of FIS and NN. The learning process of FQL is that parameters of fuzzy rules are tuned by reinforcement method generally. However, ANFIS is an approach for learning parameters of fuzzy rules by applying NN method. In this thesis, we use ANFIS to achieve an effective learning when it is hard to learn an optimal behavior by using FQL alone. There are two types of FIS in an integrated fuzzy logic system, such as FIS of FQL and FIS of ANFIS. Fuzzy rules of FQL are expressed as the partial information regarding an uncertain/unknown environment in integrated system, and a learned behavior by FQL is handled at ANFIS based on IF-THEN rules.

II. PROBLEM DESCRIPTIONS AND SURVEYS

2.1. Problem Descriptions

2.1.1. Reasoning of Human Behaviors

Firstly, we briefly mention the human behavior to clarify the technical usage of the relevant notions. In general, human behavior refer to the range of behavior caused by humans and there are a lot factors influencing human behaviors such as, culture, genetics, emotion, belief, etc. Actually, there is a difference among each person even if they are in the same circumstance. However, behavior is usually considered as a process for achieving the specific purpose. This process is no exception for animals as well as humans but it is not the same.

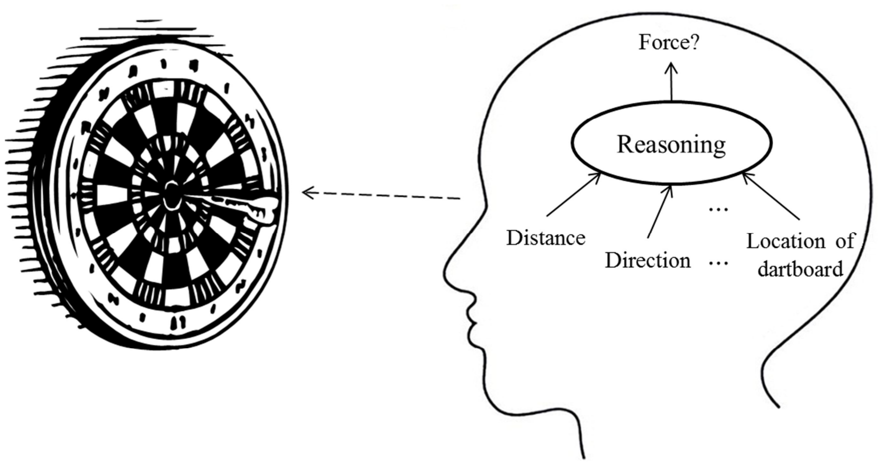


Figure 2.1.1: Human reasoning in the game of darts.

More specifically, humans are able to use reasoning process for making a decision. For instance, when humans are playing the game of darts, they consider a lot of things before shooting such as, the distance between the dartboard and their location, the direction between dartboard and their location, etc. to get a high score as shown in Figure 2.1.1. At the next shooting, they adjust the shooting action for better result based on previous experience. It means that humans have predicted a certain action in advance by using reasoning process, and they are able to learn an optimal behavior achieving their purpose by adjusting their behavior. We deal with this kind of pattern of human behavior. Our goal is that suggesting a learning method that intelligent agents perform like humans by mimicking a process of human reasoning as the game of darts. Intelligent agents are able to predict a certain action based on *a priori knowledge* in partially known environment, and learn an optimal behavior through trial and error.

2.1.2. Problems Descriptions

We recap that previous study [Lee, SW 2008] dealt with the learning algorithm which can learn patterns of human behavior by classifying the measured pattern data automatically in the uncertain/unknown environment. The studied learning algorithm is able to discover human behavioral actions and predict which actions would be taken for assisting people with disabilities. In this thesis, we deal with a learning method that learns human reasoning by using partially known information. As we already have mentioned briefly, most of the patterns of human behavior based on a process of reasoning involve ambiguity, complexity, and inconsistency. So, it is not easy or highly complex to model as the mathematical equation. On the other hand, fuzzy logic is able to describe a process of reasoning as linguistic words in intuition of human reasoning. We revisit the game of darts in the previous section. Human have made a decision before shooting by considering the environment and situation. We can describe a process of human reasoning as linguistic words for Example 1, assuming that a relation between the distance between the dartboard and the position of gamer and the strength level is only considered.

Example 1: *The description of linguistic words regarding human reasoning in the game of darts.*
(1) **IF** *the distance* is *close* **THEN** *the strength level* is *weak.*
(2) **IF** *the distance* is *far* **THEN** *the strength level* is *strong.*

Note that Example 1 shows that fuzzy rules are usually based on making a decision of humans by describing IF-THEN rules. There are two parts for representing the fuzzy rule such as, IF part and THEN part. IF part of each fuzzy rule is called a precondition part and THEN part is called a conclusion part. It is more convenient way to design the systems rather than analytical task models. Thus, we can intuitively know that the fuzzy logic is very powerful method to deal with uncertain, ambiguous, and inconsistent problem. However, extracted knowledge expressed by linguistic words needs to adjust for performing a better result. We recall the game of darts in the previous section again. After shooting the dart pin, humans adjust their behavior depending on whether or not the dart pin hits the target that they aimed. Indeed, there might be differences between predicted behavior and desired behavior. So, the precondition part and/or the conclusion part are/is needed to tune for finding desired behavior. For these reasons, a lot of ways have studied to adjust FIS automatically. One of them, we deal with the FQL method, which tunes FIS by reinforcement method, mainly in this thesis.

A lot of types of reinforcement learning methods are proposed for finding an optimal strategy in an uncertain/unknown environment such as TD method, Q-Learning, SARSA, etc. [Sutton, RS 1998; Sutton, RS 1988; Watkins, CJCH 1989; Rummery, GA 1994]. Indeed, reinforcement learning method has been successfully applied to various fields such as, control, finance, games, etc. [Yang, Q 2008;

Moody, J 2001; Wender, S 2008]. Q-Learning, which is one of reinforcement learning methods, is an attempt to build a Q-Table that is learning memory data to store estimating the discounted future rewards about taking actions from given states. A size of Q-Table is denoted by the multiplication of a length of states and a length of actions. These environment variables such as, states and actions are generally continuous in real world. In case that a problem has been encountered as the size of Q-Table is infinite. So, these continuous variables need to be converted into discrete space to build the reasonable number of the state-action pairs. This property is possible to reduce the size of Q-Table but a learned optimal behavior of agent might be unnatural. On the other hand, it needs a lot of time to find an optimal behavior in case that the resolution of discrete variables is high since the size of Q-Table has been exponentially increased. Many kinds of methods were proposed to solve this problem such as CMAC, FQL, SOM to reinforcement learning, etc. [Sutton, RS 1996; Kohone, T 1982; Smith, AJ 2002; Jouffe, L 1998; Gloremmec, PY 1997]. Especially, FQL uses *a priori knowledge* that is extracted by human reasoning during a process of learning. In addition, this extracted knowledge allows the agent to be able to reduce the time of learning by applying a function approximator. Also, the agent is possible to act continuous behaviors by referring the fuzzy rules regarding given states.

A learning structure of FQL is that FIS are adjusted by reinforcement learning method. As we have already shown in Example 1, FIS consists of two parts such as, the precondition part and the conclusion part. It means that there are three ways to adjust FIS as tuning the precondition part, tuning the conclusion part, and tuning both parts. In general, there are two phases for FIS learning, structural learning and parametric learning. The first phase is considered as changing the number of linguistic words. For instance, the precondition part of Example 1 is able to be changed as Example 2.

Example 2: *The structural learning of FIS*

(1) **IF** *the distance* is *close* and *the weight of dart pin* is *light* **THEN** *the strength level* is *weak.*

(2) **IF** *the distance* is *far* and *the weight of dart pin* is *heavy* **THEN** *the strength level* is *strong.*

Note that Example 2 shows that there are two linguistic variables in the precondition part of FIS. The second phase is considered as changing the shape of membership function of linguistic words. For example, we are able to consider four cases as Example 3 regarding the distance in the game of darts when the range of the distance is bounded from zero to zero-point-five meters.

Example 3: *The parametric learning of FIS*

(1) *The distance between the dartboard and the position of gamer is 0.5 meters.*

(2) *The distance between the dartboard and the position of gamer is 1 meter.*

(3) *The distance between the dartboard and the position of gamer is 2 meters.*

(4) *The distance between the dartboard and the position of gamer is 5 meter.*

Intuitively, case (3) and (4) of Example 3 is easily considered as *long* and case (1) of Example 3 is *close* when the number of linguistic variables of the distance is two, *close* and *long*. However, case (2) of Example 3 is ambiguous. This ambiguity is determined as the membership function of linguistic word and the shape of the membership function depends on values of parameters. The phase of parametric learning adjusts the parametric values of membership function. Structural learning and parametric learning are causing the problems that the system becomes large and complex, and the number of the degree of the freedom is increased [Jouffe, L 1998]. For these reason, the conclusion part is only tuned in FQL method.

Generally, the conclusion part that is tuned by reinforcement learning method consists of singleton values of available actions at given states in FQL method. Because of this property, it is hard to expect the efficient learning when the behavioral strategy that corresponds to the learning objectives is sole, and the number of singleton values is small. Indeed, Behaviors of the agent depend on the number of singleton values even if the agent is possible to act continuous behaviors through FQL. We can expect to learn the optimal behavior at given states by increasing the number of singleton values in the conclusion part of FIS, whereas it need a lot of time to find the optimal behavior because of increasing amount of learning data. Needing a long time for learning is also considered as inefficient learning.

ANFIS that was proposed by Jang is one of the learning methods to find a nonlinear model at given input-output pairs [Jang, J-SR 1991; Jang, J-SR 1993]. Indeed, this learning method has been successfully applied to various fields such as, control, classification, and prediction problems [Efendigil, T 2009]. ANFIS is a combination version of FIS and NN. The learning structure of ANFIS is that parameters of FIS are tuned by NN, forward learning and backward learning. ANFIS also uses *a priori knowledge* for learning. The agent searches the behavioral strategies through FQL based on q-values for each fuzzy rules. On the other hand, FIS of ANFIS is adapted to perform the desired output at given input by tuning the precondition part and the conclusion part. Because of this property, ANFIS is able to find the behavioral strategy that corresponds to the learning objectives even if the behavioral strategy is sole. However, problems are able to come from complexity of system when the dimensionality of observed states is high. It causes plenty of computation and increasing the number of learning parameters. Especially, the output of ANFIS is very sensitive depending on changing learning parameters of each state.

For these reason, we suggest an integrated fuzzy logic method which combines FQL with ANFIS. We can expect two improvements regarding the problem of learning efficiency caused by singleton values of FQL and the problem of sensitive output caused by increased learning parameters of ANFIS. More specifically, the proposed method first makes a decision based on partially known information. The complex environment is considered as the number of behavioral strategies at observed states.

Under the decision making, ANFIS finds the desired action. A numerical value of decision making is applied to the input of ANFIS and the output of ANFIS is considered as actual action which is performed by the agent. Parameters of FQL and ANFIS are adjusted by reinforcement signal returned by an environment regarding actual output and error signal different between desired output and actual output, respectively.

2.2. Surveys

2.2.1. Fuzzy Inference System

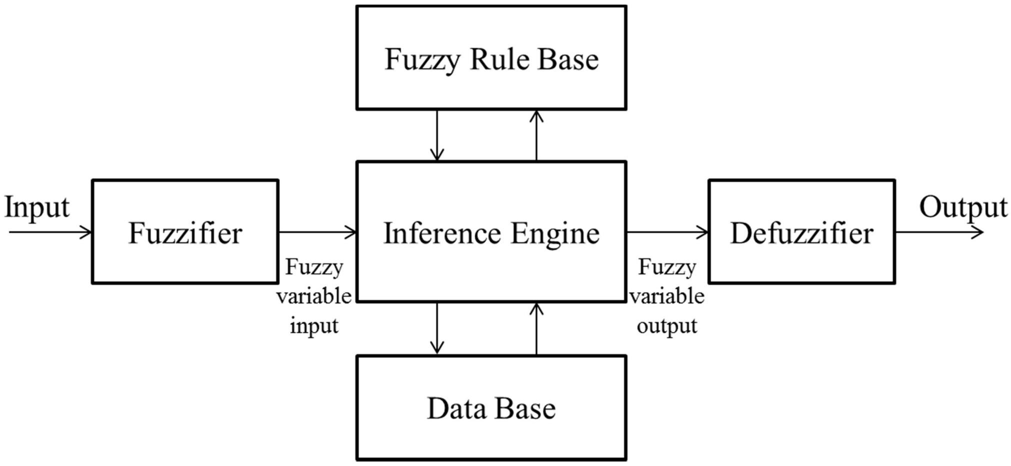


Figure 2.2.1: Structure of a Fuzzy Rule-based System.

FIS is a system that uses fuzzy logic for a process of mapping from given input data to output data. The system can be roughly separated by 3 parts such as, fuzzification, inference, and diffuzification as shown in Figure 2.2.1.In general, FIS deals with fuzzy variables instead of crisp variables. However, most of the controllers deal with crisp values. So, it is necessary for operators that the crisp variables are converted to fuzzy variables, and fuzzy variables are converted to crisp variables when FIS is used as a controller. These processes are possible to be operated by fuzzifier and defuzzifier, respectively. The inference process is composed of three functional operators such as, fuzzy rule base, inference engine, and data base. The fuzzy rule base consists of IF-THEN rules. A process of reasoning is described as linguistic words in fuzzy rule base. The data base contains the membership functions of a set of fuzzy variable that is used in fuzzy rule base. The inference engine performs the inference operation for outputs of fuzzy variables at given inputs of fuzzy variables. In summary, crisp inputs are converted to input fuzzy variables by fuzzifier. Output fuzzy variables are inferred by a process of inference. It depends on fuzzy rule base and the membership function of fuzzy rules. Finally, output fuzzy variables are converted to crisp outputs by defuzzifier.

A. General Description of fuzzy rules

The general form of fuzzy rule of fuzzy rule base is described as Equation (2.1).

$$\boldsymbol{R_i}\text{: }\mathbf{IF}\ x_1 \text{ is } L^i_{x_1} \text{ and } x_2 \text{ is } L^i_{x_2} \text{ and} \ldots \text{and } x_n \text{is } L^i_{x_n}$$
$$\mathbf{THEN}\ y_1 \text{ is } L^i_{y_1},\ \ y_2 \text{ is } L^i_{y_2},\ \ \ldots y_m \text{ is } L^i_{y_m} \tag{2.1}$$

In Equation (2.1), means of each character are as in the following:

R_i i-th fuzzy rule,

x n-dimensional input vector,

$L_{x_n}^{i}$ the linguistic words regarding n-th input variable in i-th fuzzy rule,

y m- dimensional output vector,

$L_{y_m}^{i}$ the linguistic words regarding m-th output variable in i-th fuzzy rule.

Note that Equation (2.1) shows MIMO system, dimensionality of inputs and outputs are equal to n and m, respectively. The membership degrees regarding each input variable are computed by membership function when the n-dim input vector comes to input interface of FIS, and these membership degrees are used to calculate the truth value of i-th fuzzy rule. Finally, the outputs are computed based on truth values corresponding fuzzy rule. These processes are performed by the precondition part (IF part) and the conclusion part (THEN part), respectively.

B. Precondition Part

The membership degrees regarding crisp input variables are determined by member functions in the precondition part. There are various kinds of fuzzy membership functions such as triangular, trapezoidal, Bell, Gaussian, and Sigmoidal shapes, and these shapes are generally convex forms. We recall the fuzzy variable, *distance,* which has already been mentioned in the game of darts of Example 2 for more details. Suppose that a set of linguistic words consists of four linguistic labels such as, *very close, close, far,* and *very far* then the fuzzy variable is possible to be expressed as shown in Figure 2.2.2.

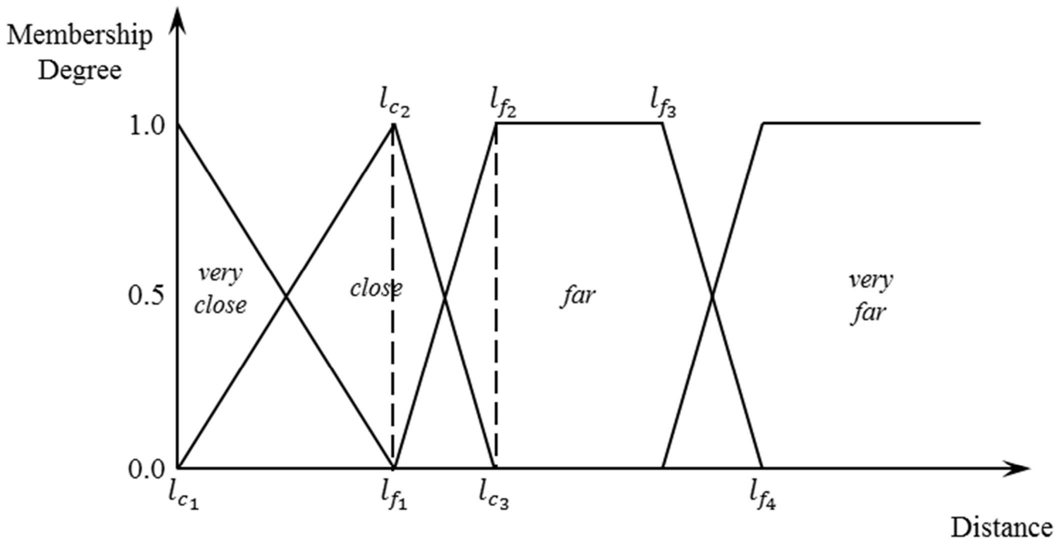


Figure 2.2.2: Strong fuzzy partition of a fuzzy variable for *distance*.

Note that Figure 2.2.2 shows a fuzzy variable which consists of triangular and trapezoidal shapes. Shapes of membership functions are depending on parameters that are associated with membership function. The membership shapes of *close* are able to change by tuning parameters such as, l_{c_1}, l_{c_2}, and l_{c_3}. In addition, we can easily know that l_{c_2} which is one of parameters of *close* and l_{f_1} which is one of parameters of *far* are equal. It means that only two linguistic words are overlapped in fuzzy rules. So, at least one of the fuzzy set is activated and there are no three fuzzy sets which are activated. As Equation (2.2), the sum of membership degrees regarding a given n-th input value x_n is one. This fuzzy partition which has these properties is called strong fuzzy partition.

$$\sum_{i=1}^{N_R} \mu_{L_{x_n}}^{i}(x_n) = 1 \tag{2.2}$$

In Equation (2.2),

$\mathrm{N_R}$ the total number of fuzzy rules,

$\mu_{L_{x_n}^i}$ a membership degree regarding a given input value x_n of i-th fuzzy rule.

The membership degree of a given input value x_n for fuzzy label *close* in Figure 2.2.2 is computed by following Equation (2.3).

$$\mu_{L_{x_n}}^{i}(x_n) = \begin{cases} 0.0, & x_n < l_{c_1} \\ \dfrac{x_n - l_{c_1}}{l_{c_2} - l_{c_1}}, & l_{c_1} \le x_n < l_{c_2} \\ 1 + \dfrac{x_n - l_{c_2}}{l_{c_2} - l_{c_3}}, & l_{c_2} \le x_n < l_{c_3} \\ 1.0, & otherwise \end{cases} \tag{2.3}$$

C. Conclusion Part

The number of degrees of n is calculated in each fuzzy rule for given n-dim input vector, and each fuzzy rule has the number of output linguistic words of m as Equation (2.1). There are two types of computation methods such as, min-max (AND-OR) and max-min (OR-AND). Min and max are mathematically operated as T-norm and S-norm, respectively. We are focusing on min-max method in this thesis. Firstly, a truth value of each fuzzy rule is calculated by T-norm of n membership degrees as Equation (2.4).

$$v_i(x) = T\left(\mu_{L_{x_1}}^{i}(x_1), \mu_{L_{x_2}}^{i}(x_2), \ldots, \mu_{L_{x_n}}^{i}(x_n)\right) \tag{2.4}$$

Linguistic words of output variable in each fuzzy rule are fuzzy variables or crisp variables. These output variables are projected by the truth value of each fuzzy rule. The number of output fuzzy variable of m is computed by S-norm operation. We here give two types of typical FIS models such as Mamdani and TSK models.

D. Mamdani type Fuzzy Inference System

A Mamdani type FIS is widely used to extract expert knowledge, linguistic words of this system consists of fuzzy variables in the precondition part and the conclusion part. More specifically, we give an example for FIS with MISO as Example 4.

Example 4: *Fuzzy inference system with MISO for two fuzzy rules.*

R_1: **IF** x is X_1 and y is Y_2 **THEN** z is Z_1

R_2: **IF** x is X_2 and y is Y_1 **THEN** z is Z_2

Example 4 shows the two inputs and a single output FIS with two fuzzy rules. We assume that each fuzzy variable has two linguistic words and its membership shapes are triangular. For instance, the membership shapes of inputs and output fuzzy linguistic words are shown in Figure 2.2.3.

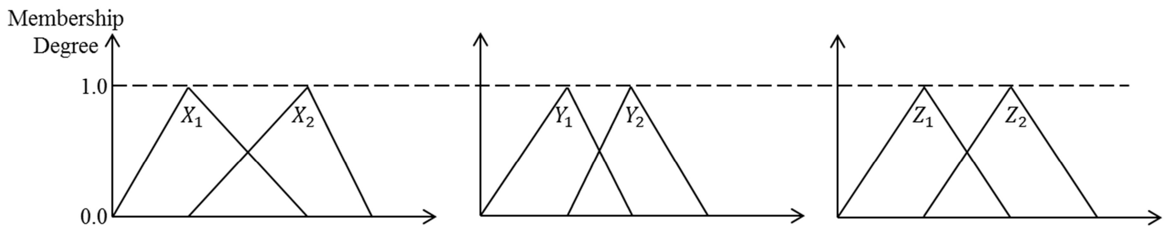


Figure 2.2.3: Membership function of linguistic words in Example 4.

The computational procedure of Mamdani type FIS, which is based on fuzzy rules of Example 4 and membership functions of Figure 2.2.3, is shown in Figure 2.2.4. There are several processes to compute the output of this FIS when the crisp inputs x and y come to input interface. The purpose of fuzzifier is to map the crisp inputs such as measured sensor data to fuzzy variable by using membership functions. It means that the membership degrees of inputs are computed regarding linguistic words. The truth value of each fuzzy rule is calculated from membership degrees by T-norm operation, and the fuzzy variable of the conclusion part is project by truth value. Black trapezoids of THEN part in Figure 2.2.4 is a clipped output membership function. The fuzzy output is expressed by S-norm operation of clipped output membership functions of each fuzzy rule. A single crisp output from the Mamdani type FIS is computed by defuzzification of fuzzy output. There many types of defuzzification methods such as, center of gravity, center of sum, center of largest area, etc. For instance, center of gravity method is shown in Figure 2.2.4.

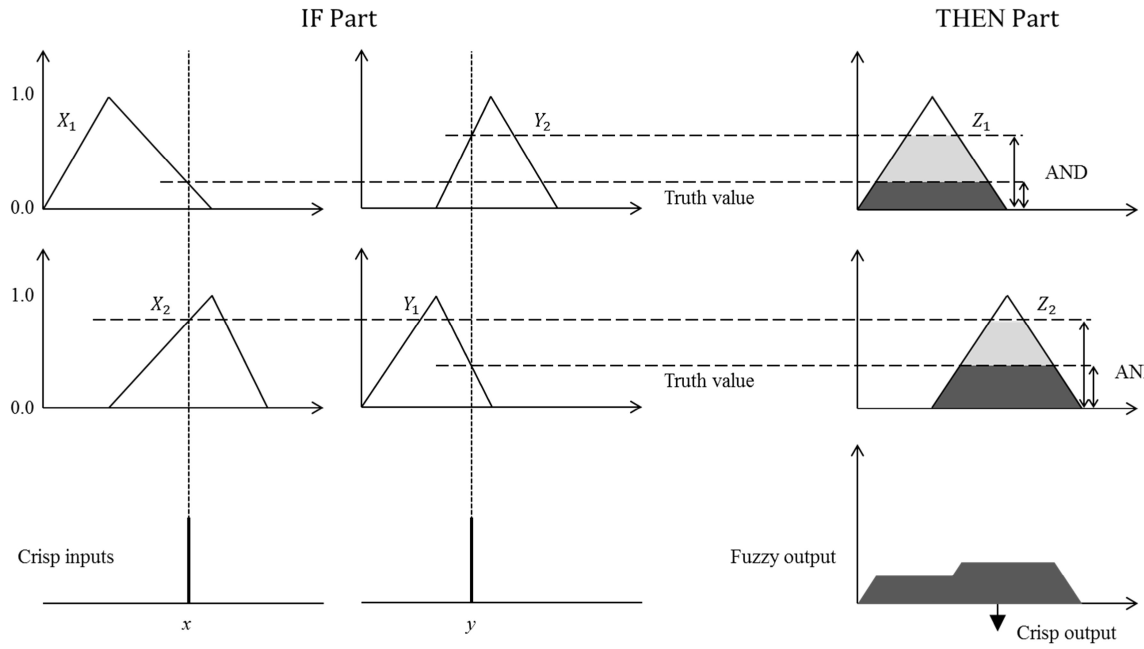


Figure 2.2.4: A Mamdani type FIS with two inputs and a single output.

E. Takagi-Sugeno-Kang type Fuzzy Inference System

A TSK type FIS is similar with Mamdaini type FIS which has been mentioned previous section. However, the most difference between TSK type and Mamdani type is that the conclusion part of TSK type consists of a set of *n* order equations as Equation (2.5). It means that the outputs of Mamdani type FIS are computed by defuzzification of fuzzy outputs that are calculated by S-norm operation of clipped output membership functions.

$$\boldsymbol{R_i}\text{: }\textbf{IF } x_1 \text{ is } L^i_{x_1} \text{ and } x_2 \text{ is } L^i_{x_2} \text{ and ... and } x_n \text{is } L^i_{x_n}$$
$$\textbf{THEN } y_1 = f(x_1, x_2, \dots, x_n), y_2 = f(x_1, x_2, \dots, x_n), \dots, y_m = f(x_1, x_2, \dots, x_n) \quad (2.5)$$

In case of TSK type, crisp outputs are computed by weighted sum of outputs from *n* order equation in each fuzzy rule. More specifically, we revisit Example 4 in previous sections which is two inputs and a single output FIS. We use the same precondition part and linguistic words as Example 4 and Figure 2.2.3, respectively. On the other hand, the conclusion part of fuzzy rules consists of crisp functions as Example 5 instead of fuzzy variables.

Example 5: *TSK type FIS with MISO for two fuzzy rules*

R_1: **IF** x is X_1 and y is Y_2 **THEN** $z = x - 3y$

R_2: **IF** x is X_2 and y is Y_1 **THEN** $z = 2x + \mathrm{y}$

Note that Example 5 shows that two inputs and a single output TSK type FIS with two fuzzy rules.

We assume that the conclusion part of FIS is a crisp functions which consists of input values. The computational procedure of TKS type FIS is similar with Mamdani type FIS. However, the crisp output of system is computed by weight sum of truth values and the output of crisp functions. It means that the truth value is considered as the ratio of each fuzzy rule output. An example of TKS type FIS is shown in Figure 2.2.5.

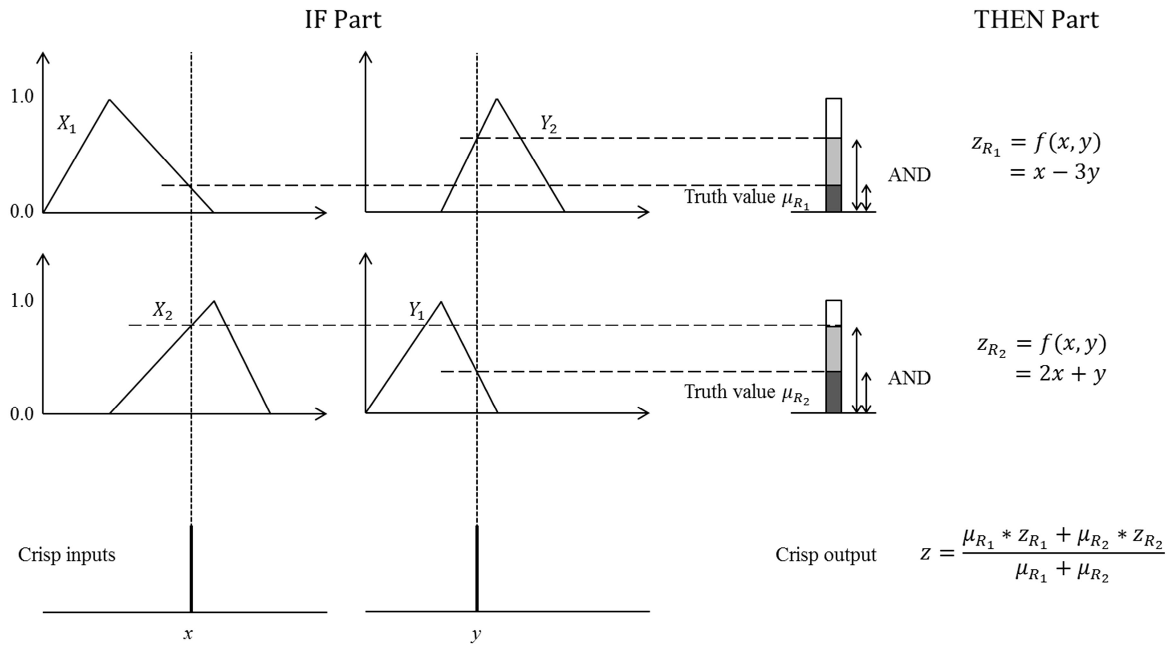


Figure 2.2.5: A TSK type FIS with two inputs and a single output.

We review the FIS in this section and briefly mention the two types FIS, Mamdani type and TSK type models with simple examples. Mamdani type FIS consists of fuzzy variables in both parts. On the other hand, TSK type FIS consists of fuzzy variables in the precondition part and crisp function in the conclusion part. These two fuzzy models use IF-THEN rules that are based on extracted knowledge from experts or humans to infer crisp outputs for given crisp inputs.

FQL and ANFIS methods use FIS for learning. Ordinary FIS has several processes to compute crisp outputs such as fuzzification, inference, and defuzzification. However, FIS is tuned by reinforcement learning method and neural network method in FQL and ANFIS, respectively to find an optimal behavior. Also, these methods use the partially known information which is expressed as IF-THEN rules of fuzzy rules based on a process of human reasoning.

2.2.2. Q-Learning

Q-Learning is one of reinforcement learning methods, and it performs an on-line learning to find an optimal behavior in an uncertain/known environment. It means that the agent is able to have the capability of learning to act optimally in an uncertain/unknown environment by experiencing the consequences of actions even if there are no pre-planned behaviors. So, the purpose of Q-Learning method is to estimate the Q-values for an optimal strategy. Q-values are expressed as the sum of the discounted future rewards for taking actions from given states and following an optimal strategy. The equation of updating Q-value is denoted as Equation (2.6).

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q^*(s') - Q(s,a)] \tag{2.6}$$

$$Q^*(s') = \max_{a \in A(s')} Q(s',a)$$

In Equation (2.6),

$\mathrm{Q(s,a)}$ a Q-value for taking action a from given states s,

α a learning rate of Q-Learning,

r a reinforcement signal received from environment for taking action a from given state s,

γ a discount rate of Q-Learning,

$Q^*(s')$ a maximum Q-value at a new state s' taken by action a,

$A(s)$ a set of actions for given states s.

The learning rate α determines a degree of the newly learned value which will override the current Q-value. When the learning rate is equal to zero, it means that the agent does not learn anything. In case that the learning rate is one, the agent focuses on the recent information. The discounting rate γ determines a degree of future rewards. When the value of discounting rate is zero, the agent is only considering current reward. On the other hand, when discounting rate is close to one, the agent is considering long-term rewards.

The agent moves from given state s to a new state s' and receive a reinforcement signal r from an uncertain/unknown environment when the agent has chosen action a from given states s. The updating equation of Q-value is considered as the sum of current Q-value and error signal regarding a new state s' when $r + \gamma Q^*(s')$ is learned value. More specifically, the procedure of Q-Learning algorithm is following as below:

- observes current state s of the agent and choose and take action a,
- observes a new state s' caused by action a and receives a reinforcement signal r,
- updates Q-value following Equation (2.6)

· replaces current state s with s'

· repeats these procedures for satisfying the stop condition.

These computational procedures are called an episode. The agent escapes the episode when the observed states are satisfying a goal state. The agent usually stores Q-values in a look-up table when the agent visits every state-action pair continuously. Q-Learning is ensured to converge to the correct Q-values when an uncertain/unknown environment is stationary and depends on state-action pair. It is called Markov property. In general, the look-up table is a matrix for a length of states by a length of actions. The agent selects an optimal action depending on the Q-value. Thus, the time of learning is very sensitive to the size of the look-up table of Q-Learning.

2.2.3. Fuzzy Q-Learning

FQL learning method has been developed based on Watkins' Q-Learning which reviewed in section 2.2.2. It easily knows that FQL is a combination version of FIS and Q-Learning to solve the problem of ordinary Q-Learning. We briefly mention the size of Q-table in section 2.1.2 and section 2.2.2. In general, Q-values have been stored in Q-table based on experienced state-action pair. However, problems come to a case of large state-action spaces or continuous state-action spaces. These cases are impracticable or impossible to find an optimal behavior. To solve these problems, a lot of methods were proposed to reduce and store Q-values such as NN, SOM. However, these approaches have caused slow processes. Another approach combines Q-Learning with fuzzy logic here. Generally, the fuzzy rules are described as IF-THEN rules that are based on a process of human reasoning. It is possible to express an uncertain/unknown environment partially to help the learning. It means that the agent is able to predict an appropriate action from given states by using partially known information which is described as fuzzy rules. Moreover, the fuzzy logic is applied to the function approximator to reduce the size of a look-up table as well as the agent can act continuous actions by using truth values regarding given states.

A. Representation of fuzzy inference system

We have briefly mentioned the ordinary FIS in section 2.2.1. The general form of FIS is expressed as Equation (2.1). The linguistic words in the precondition part and the conclusion part might be fuzzy variables or crisp variables. However, there are little differences in FIS of FQL. It is represented as Equation (2.7).

$$\begin{aligned} &R_i\text{: } \mathbf{IF}\ s_1 \text{ is } L^i_{s_1} \text{ and } s_2 \text{ is } L^i_{s_2} \text{ and ... and } s_n \text{ is } L^i_{s_n} \\ &\qquad \mathbf{THEN}\ a_{i,1} \text{ with } q_{i,1} \text{ or } a_{i,2} \text{ with } q_{i,2} \text{ or ... } a_{i,j} \text{ with } q_{i,j} \end{aligned} \tag{2.7}$$

In Equation (2.7),

R_i — i-th rule of fuzzy rule base,

s — n-dimensional input state vector observed from sensors

$L^i_{s_n}$ — the linguistic word of n-th input variable in fuzzy rule R_i,

$a_{i,j}$ — j-th action in fuzzy rule R_i,

$q_{i,j}$ — a local singleton q-value corresponding j-th action in fuzzy rule R_i.

Note that Equation (2.7) shows a representation of FIS of FQL. The precondition part of FIS is similar with the ordinary FIS. Linguistic words are expressed as membership functions such as triangular, trapezoidal, Bell, Gaussian, and Sigmoidal shapes, and the truth value of fuzzy rule R_i is computed by T-norm operation. On the other hand, the conclusion part of FIS consists of a set of available actions and a set of local singleton q-values which is corresponding each action instead of fuzzy variables. In FQL method, the number of linguistic words and their positions in the precondition part are set in advance based on *a priori knowledge* of experts or humans. So, the conclusion part of FIS is only tuned by reinforcement learning method. It is to prevent that the system is large and complex, and increasing the number of the degree of freedom.

B. A computational procedure of FQL

A learning structure of the precondition part is similar with ordinary FIS. The observed states are considered as crisp inputs in FQL method. Indeed, the observed states data come from various sensors in the real world. The agent is able to recognize a current situation from observed states. From Equation (2.7), the membership degrees for given state vector *s* in fuzzy rule R_i are firstly computed in the precondition part. The truth value of fuzzy rule R_i is calculated by T-norm operation of the membership degrees. The computational equation of truth value in fuzzy rule R_i is expressed as Equation (2.8) when the dimensionality of input states is *n*.

$$v_i(s) = \mathrm{T} - norm\left(\mu^i_{L_{s_1}}(s_1), \mu^i_{L_{s_2}}(s_2), \dots, \mu^i_{L_{s_n}}(s_n)\right) \tag{2.8}$$

In the conclusion part, one of available actions is selected based on corresponding local singleton q-values in fuzzy rule R_i by using EEP method from given states. Making decision of FQL method is determined by weighted sum of truth values and selected actions. This is expressed as Equation (2.9).

$$a = \frac{\sum_{i=1}^{N_R} v_i(s) * a_{i,j'}}{\sum_{i=1}^{N_R} v_i(s)} \tag{2.9}$$

In Equation (2.9),

a determined behavior from local actions of FQL,

$v_i(s)$ the truth value of *i*-th fuzzy rule,

$a_{i,j'}$ the selected local action of *i*-th fuzzy rule,

N_R the total number of fuzzy rules.

Note that Equation (2.9) shows the computational process of a global action that is determined by local actions and truth values. The truth value of each fuzzy rule is considered as the ratio of *i*-th fuzzy rule in total fuzzy rules. The agent act the global action in complex environment when the decision is made from IF-THEN rules which is described as a process of reasoning extracted from experts or humans. The environment gives a reinforcement signal *r* to the agent as the result of taken global action at given states. This signal might be a positive real number (reward) or a negative real number (punishment). The local q-value in the conclusion part of each fuzzy rule is tuned based on reinforcement signal. The updating equation of FQL uses the same equation of ordinary Q-Learning. It is denoted as Equation (2.10).

$$q_{i,j'} = q_{i,j'} + \Delta Q(s,a)e_{i,j'} \tag{2.10}$$

$$\Delta Q(s,a) = \alpha\big(r + \gamma Q^*(s) - Q(s,a)\big)$$

$$Q(s,a) = \frac{\sum_{i=1}^{N_R} v_i(s) * q_{i,j'}}{\sum_{i=1}^{N_R} v_i(s)}$$

$$Q^*(s) = \frac{\sum_{i=1}^{N_R} v_i(s) * \max_{j^* \le j} q_{i,j^*}}{\sum_{i=1}^{N_R} v_i(s)}$$

$$e_{i,j} = \begin{cases} \lambda\gamma e_{i,j} + \dfrac{v_i(S)}{\sum_{i=1}^{N_R} v_i(S)} & if\ j = j' \\ \lambda\gamma e_{i,j} & otherwise \end{cases}$$

where,

$q_{i,j'}$ the local q-value of the selected local action for *i*-th fuzzy rule,

α a learning rate of FQL,

r a reinforcement signal received from environment as the result of taken global action *a* at given states *s*,

γ a discount rate of Q-Learning,

$Q(s,a)$ a global Q-value denoted by weighted sum of the local q-values of the selected local actions and truth values of total fuzzy rules,

$Q^*(s)$ a value function for given states *s* denoted by weighted sum of the maximum q-value of each fuzzy rule and truth values,

$e_{i,j}$ the eligibility of an actions to speed up learning,

λ a parameter of eligibility trace.

Note that learning factors, such as α, γ, and *r* are the same factors with ordinary Q-Learning that has already been mentioned in section 2.2.2. The parameter λ refers to the use of the eligibility trace method to speed up learning. It is equal to ordinary Q-Learning when λ is zero. The local q-value of selected local action in fuzzy rule R_i is updated as much as the ratio of truth value in rule R_i from total fuzzy rules. More specifically, the procedure of FQL method is following as below:

· computes membership degrees regarding observed input state vector *s*,
· computes truth values through T-norm operation of membership degrees,

· selects local actions based on corresponding local q-values by using EEP method,

· determines a global action *a* as Equation (2.9),

· receives a reinforcement signal *r* for taking action *a* at observed input state vector *s*,

· updates local q-values as following Equation (2.10),

· repeats these procedures for satisfying the stop condition.

The local singleton q-values in the conclusion part of FIS are adjusted for finding an optimal behavior. The agent is able to act a continuous action and reduce the size of a look-up table by combining FIS with Q-Learning. The agent repeats learning procedure of FQL when a behavior of the agent is satisfying the stop condition. The size of a look-up table is denoted as multiplication of the total number of fuzzy rules and the length of actions. The adjusted local q-values are usually stored in a look-up table. Also, FQL method ensures the convergence to find the correct local q-values when the environment is stationary and depends on observed state-global action pair.

2.2.4. Adaptive Network based Fuzzy Inference System.

The ANFIS proposed by Jang is a learning method which integrated FIS and NN. This method is an approach to model a nonlinear operator based on input-output pairs and an extracted knowledge process from experts or humans. In general, system modeling based on differential equation is hard works or unsuitable regarding an uncertain system. In this case, an approach of the fuzzy logic system is possible to deal with the uncertain system modeling easily by employing IF-THEN rules which is based on a process of human reasoning. However, there might be differences between the extracted knowledge and a real system. So, it needs to adjust the fuzzy logic system. The ANFIS is one of intelligent learning method to adjust the FIS automatically based on NN method. In the FQL which has already reviewed in section 2.2.3, the conclusion part of FIS is only tuned by reinforcement learning method. On the other hand, the precondition part and the conclusion part of FIS are tuned by NN method in ANFIS. The parameters of both parts are adjusted and the number of linguistic words depends on the extracted knowledge. ANFIS uses the TSK type fuzzy model. As briefly mentioned, TSK type fuzzy model consists of fuzzy variables and crisp function equations. There are two phases for learning structures such as, forward learning and backward learning methods. The coefficients of crisp function are tuned by forward learning process and the parameters of membership function are tuned by backward learning process. It is called hybrid learning method.

A. A system architecture of ANFIS

We have briefly reviewed the TSK type fuzzy model as Equation (2.5). More specifically, we give an example for first order TSK type fuzzy model with two inputs and a single output as Example 6.

Example 6: *First order TSK type fuzzy model.*

R_1: $\textbf{IF}\ x\ is\ X_1\ and\ y\ is\ Y_1$ $\quad\textbf{THEN}\ f_1(x,y) = p_1x + q_1y + r_1$

R_2: $\textbf{IF}\ x\ is\ X_2\ and\ y\ is\ Y_2$ $\quad\textbf{THEN}\ f_2(x,y) = p_2x + q_2y + r_2$

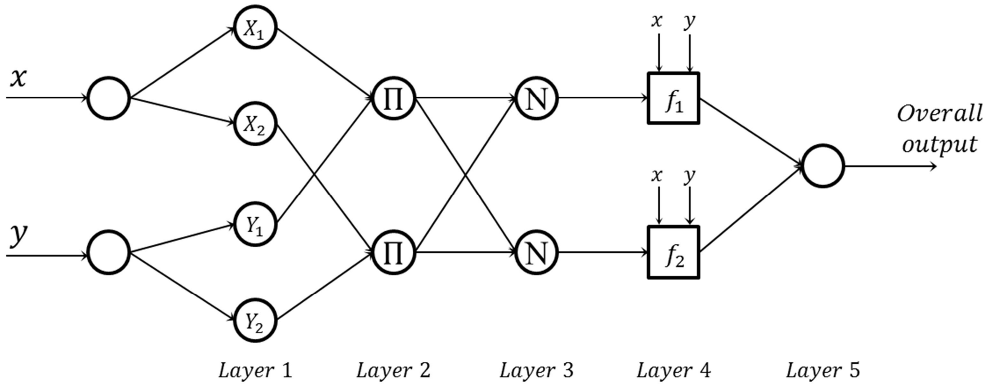


Figure 2.2.6: The system architecture of ANFIS.

Note that Example 6 shows that first order TSK type fuzzy model. The system architecture of ANFIS based on Example 6 is shown as Figure 2.2.6. There are five layers in the process of ANFIS. Each layer is considered as a function operator to compute the overall output.

Layer 1: Every node in this layer, the membership degree of node *i* is calculated by membership function of linguistic word where the input comes to node *i*. The membership shapes of linguistic words could be triangular, trapezoidal, bell shapes, etc. It is similar with ordinary FIS. Each membership function consists of a set of parameters which determine the shape of function. It is called premise parameters.

Layer 2: Every node in this layer, the truth value of node *i* is computed by T-norm operation. The inputs of layer 2 are considered as associated with membership degrees that are calculated in previous layer. It computation process is denoted as Equation (2.4).

Layer 3: Every node in this layer, the normalized truth value is computed as Equation (2.11).

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^{N_R} w_i} \tag{2.11}$$

In Equation (2.11),

w_i the truth value of node *i,*

N_R the total number of fuzzy rules.

The total number of truth values is equal to N_R. The normalized truth value means the ratio of node *i* to total node.

Layer 4: Every node in this layer, the output of node *i* is computed as Equation (2.12).

$$o_i = \bar{w}_i f_i(x, y) = \bar{w}_i(p_i x + q_i y + r_i) \tag{2.12}$$

The node output is a multiplication of the normalized truth value and the value of crisp function, and the value of crisp function is a linear combination of inputs and coefficients, Here, p_i, q_i, and r_i are considered as a set of parameters for the conclusion part. It is called consequent parameters.

Layer 5: Every node in this layer, the overall output of ANFIS is computed as Equation (2.13).

$$O = \sum_{i=1}^{N_R} \bar{w}_i f_i = \frac{\sum_{i=1}^{N_R} w_i f_i(x, y)}{\sum_{i=1}^{N_R} w_i} \tag{2.13}$$

From the Equation (2.13), the overall output of ANFIS is considered as weighted sum of node outputs and normalized truth values.

B. Hybrid learning algorithm

The two sets of parameters are represented in previous section *B* such as premise parameters and consequent parameters. These parameters are adjusted by hybrid learning method which is a combination of LSE and back propagation methods. More specifically, consequent parameters in the conclusion part are identified by the LSE method in the forward pass learning, and premise parameters in the precondition part are fixed. On the other hand, premise parameters are identified by the gradient descent method by propagating the error signal in backward pass. This is summarized in Table 1.

	Forward Pass	Backward Pass
Premise parameters	Fixed	Gradient Descent
Consequent parameters	Least Square Estimator	Fixed
Signals	Node outputs	Error signals

Table 3: The hybrid learning procedure for ANFIS

From the Example 6, the overall output of ANFIS is able to rewritten as Equation (2.14).

$$O = \bar{w}_1 f_1(x, y) + \bar{w}_2 f_2(x, y)$$

$$= (\bar{w}_1 \mathrm{x})\mathrm{p}_1 + (\bar{w}_1 y)q_1 + \bar{w}_1 r_1 + (\bar{w}_2 \mathrm{x})\mathrm{p}_2 + (\bar{w}_2 y)q_2 + \bar{w}_2 r_2 \tag{2.14}$$

In Equation (2.14), consequent parameters are considered as unknown parameters to be optimized. The purpose of LSE is to optimize unknown parameters. The formula of LSE is represented as Equation (2.15).

$$\hat{X} = (A^T A)^{-1} A^T B \tag{2.15}$$

where,

$\hat{X}$ the estimated unknown parameters,

A a function matrix regarding input vector,

B training vector.

When training data set which has P entries is given and the dimension of unknown parameter is M, the dimensions of A, B, and X are P x M, P x 1, and M x 1, respectively. Equation (2.15) has to compute the matrix inverse, and it is expensive in computation when M is a positive large number. So, the sequential LSE method is employed as Equation (2.16).

$$X_{i+1} = X_i + S_{i+1}a_{i+1}\left(b_{i+1}^T - a_{i+1}^T X_i\right) \tag{2.16}$$

$$S_{i+1} = S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, i = 0, 1, \dots, P-1$$

where,

X_{i+1} the estimated unknown parameters,

S_i the covariance matrix,

a_i^T the *i*-th row vector of matrix A defined in Equation (2.15),

b_i^T the *i*-th scalar of B defined in Equation (2.15).

In the backward pass learning, the error signals are propagated in backward pass for identifying the premise parameters. The error signals of each node are denoted as Equation (2.17).

$$E_p = \sum_{k=1}^{N(L)} \left(d_k - O_{L,k}\right)^2 \tag{2.17}$$

where,

$N(L)$ the total number of nodes with L-th layer,

d_k the *k*-th component of *p*-th desired output vector,

$O_{L,k}$ the *k*-th component of actual output vector.

Note that Equation (2.17) shows an error signal different between desired outputs and actual outputs, and an overall error signal is denoted as Equation (2.18). The purpose of task is to minimize the overall error signal through tuning parameters. The basic concept of gradient method is to pass the derivative information starting from output layer to input layer. So, we need to calculate the error rate which is expressed as Equation (2.19).

$$E = \sum_{p=1}^{P} E_p \tag{2.18}$$

$$\epsilon_{l,i} = \frac{\partial E_p}{\partial O_{l,i}} = -2\left(d_i - O_{l,i}\right) \tag{2.19}$$

For the internal node *i* of layer *l*, the error rate is able to be derived by the chain rule as Equation (2.20).

$$\epsilon_{l,i} = \frac{\partial E_p}{\partial O_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial O_{l+1,m}} \cdot \frac{\partial f_{l+1,m}}{\partial O_{l,i}} = \sum_{m=1}^{N(l+1)} \epsilon_{l+1,m} \frac{\partial f_{l+1,m}}{\partial O_{l,i}} \tag{2.20}$$

$$O_{l,i} = f_{l,i}\left(x_{l-1,i}, \dots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma, \dots\right)$$

where,

α, β, γ the parameters of node *i*.

The gradient descent method is defined as the derivative of the error measure regarding each parameter. The updated algorithm is denoted as Equation (2.21) when α is a parameter of node *i* at layer *l*.

$$\frac{\partial E_p}{\partial \alpha} = \frac{\partial E_p}{\partial O_{l,i}} \cdot \frac{\partial f_{l,i}}{\partial \alpha} = \epsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha}$$

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha}$$

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{2.21}$$

where,

η a learning rate of gradient descent method.

So, a set of premise parameters and a set of consequent parameters are adjusted by gradient descent and LSE methods, respectively toward reducing the overall error signal. There are four types of hybrid learning methods because of complexities of computational processes.

a. Gradient Descent Only: The premise and consequent parameters are updated by the gradient descent only.

b. Gradient Descent and One Pass of LSE: Once the LSE method is applied in the beginning of the process to estimate the initial consequent parameters and then all parameters are updated by gradient descent.

c. Gradient Descent and LSE: This is the hybrid learning method.

d. Sequential LSE Only: The premise and consequent parameters are updated by the sequential LSE only.

III. FUZZY LOGIC SYSTEMS BASED ON PARTIALLY KNOWN ENVIRONMENT

3.1. Descriptions

We have briefly mentioned the FQL and the ANFIS leaning methods in the previous chapter. These methods use the fuzzy logic concept in learning procedures. Traditionally, the fuzzy method is well known to handle with ambiguity and uncertainty based on fuzzy rule base which is described as IF-THEN rules. Generally, fuzzy rules are designed by the extracted knowledge from experts or humans, and the extracted knowledge is described as linguistics words in fuzzy rules. This property allows that the fuzzy logic is easy to introduce *a priori knowledge*. It means that fuzzy logic system is able to describe strategies of human behaviors based on a reasoning process. So, fuzzy rules can be considered as the partially known information to represent behavior strategies regarding an uncertain/unknown environment. So, the agent becomes possible to predict which action should be taken in their situation in an uncertain/unknown environment by referring partially known information.

In this chapter, we proposed an integrated fuzzy logic system which is FQL with ANFIS for mimicking a process of human reasoning. FQL is possible to find an optimal behavior by evaluating local q-values regarding state-action pairs. In general, the conclusion part of FIS of FQL consists of a set of local singleton actions and a set of local singleton q-values in every fuzzy rule. The local action is selected by corresponding local q-values. In FQL method, each local action is considered as a decision. More specifically, we introduce examples such as, the mountain car and the navigation problems in an indoor environment. The mountain car problem which is well known problem to deal with reinforcement learning method is shown in Figure 3.1.1. The purpose of the agent, which is an under powered car, is to drive up a steep hill for reaching the goal position. The agent should depend on the steep slope to get acceleration. So, the agent should take action such as left, neutral, and right, and singleton values of all actions are able to be defined as -1, 0, and 1 respectively. The global action is computed as a continuous value between -1 and 1 based on the IF-THEN rules of fuzzy method. From the mountain car problem, we can know that the actual action value of the agent is considered as the global action that is the decision-making.

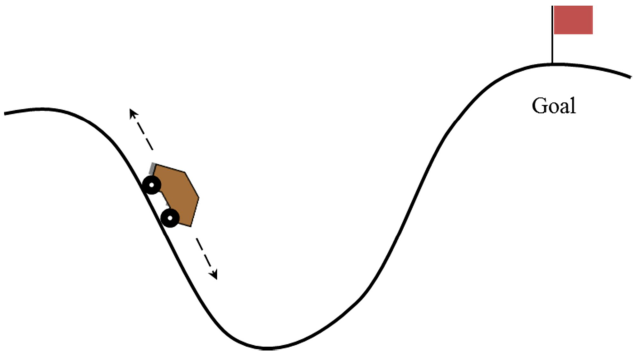


Figure 3.1.1: The mountain car problem.

Now, we consider a simple example of the navigation problem in an indoor environment that is shown in Figure 3.1.2. The indoor environment is considered as house or office. The purpose of the agent that is a navigation robot is to start an arbitrary start position and to arrive an arbitrary goal position. Suppose that there is no object except the agent in the indoor environment, and then it can be considered as a problem of the same type with the mountain car problem because the agent only needs to find an optimal path to arrive the goal position.

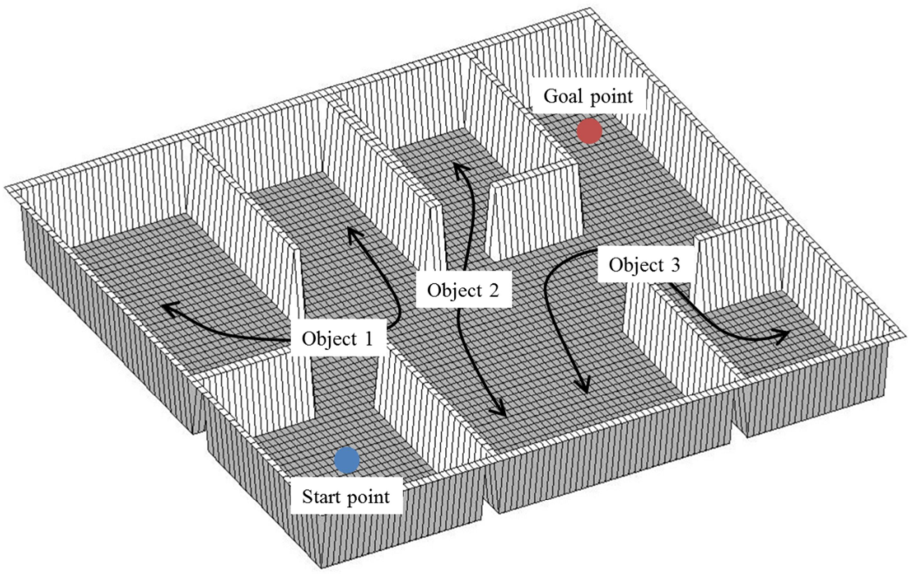


Figure 3.1.2: The simple navigation problem.

However, in case that there are other objects which move in the indoor environment freely, the agent should have learned the ways to avoid a collision between the agent and the objects. So, the agent should make decisions such as, staying, avoiding, and passing as well as finding an optimal path to reach the goal position. We assume that a set of local action in the conclusion part consists of five actions such as up, down, right, left, and stop, and three intelligent objects move the specific path repeatedly as Figure 3.1.2. In this case, it needs a lot of time for learning rather than no objects case because the agent should find an optimal path and the ways to avoid the intelligent objects which move repeatedly. So, it is hard to expect the efficient learning time. We are able to improve an efficiency of leaning by integrating fuzzy logic controller such as FIS. For instance, the local actions of the conclusion part set the staying, avoiding, and passing behaviors, and the agent makes a decision based on IF-THEN rules. The actual output values of the agent depend on the system outputs of FIS by referring decision-making. So, the agent is possible to find optimal strategies more effectively.

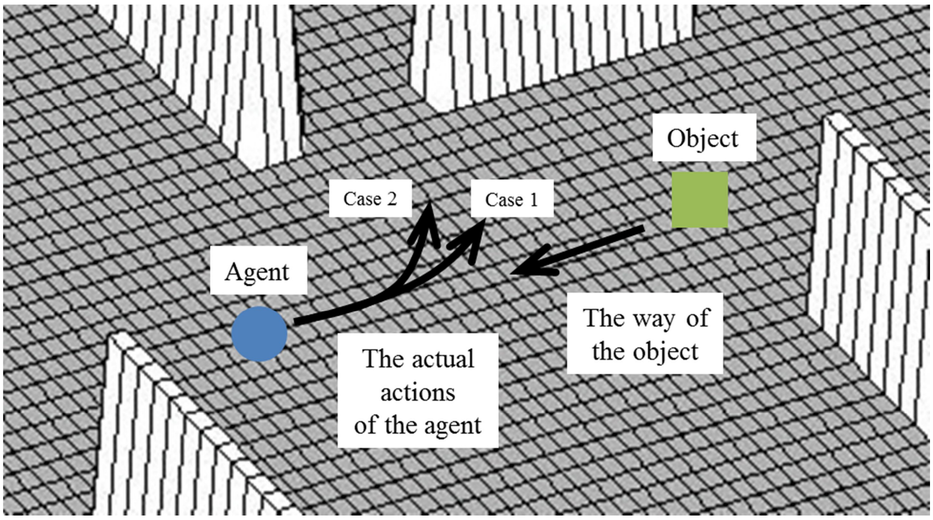


Figure 3.1.3: The avoiding actions of the agent.

We can consider a situation of the agent which is shown in Figure 3.1.3. We assume that the agent select the avoidance action to left side from given states that are considered as the measured data from ultrasonic sensors, and the actual actions, such as velocity and torque are computed by FIS to avoid the object. There are two cases in the computed outputs as Figure 3.1.3. Intuitively, the first case is more appropriate action rather than the second case. However, the second case also does not matter to avoid the object. It means that both cases don’t cause any problems for avoiding actions. On the other hand, we can consider a situation that the angle value is required with the avoidance action. It needs to adjust the parameters of FIS to satisfy the desired output values. It is more complex problem. ;.

3.2. FQL with ANFIS

A. Design of FQL

In this section, we specify two fuzzy logic systems. Firstly, we define the 0-order TSK type fuzzy model, and the formula is expressed as Equation (2.7) in section 2.2.3. The precondition part of TSK type FIS of FQL consists of fuzzy variables. In FQL method, the local singleton q-values are only updated, and the parameters of the precondition part are fixed. So we use a simple membership functions such as triangular and trapezoidal shapes instead of Bell, Gaussian, and Sigmoidal shapes. Especially, an isosceles triangular membership shape is more convenient to calculate the membership degree, and it is easy to deal with intuitive representation rather than trapezoidal membership shape. So, we use an isosceles triangular membership shape, which is shown in Figure 3.2.1, in this method.

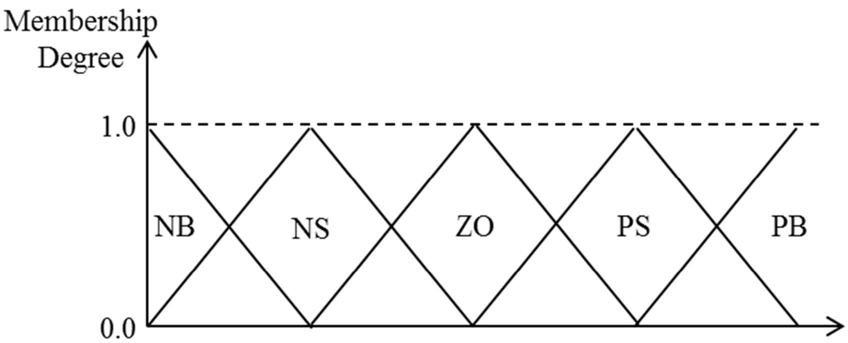


Figure 3.2.1: The isosceles triangular membership functions.

Note that Figure 3.2.1 shows that the fuzzy set which has five linguistic words such as NB, NS, ZO, PS, and PB, and we also use a strong fuzzy partition that is defined as Equation (2.2). The membership degrees regarding the observed states are computed as Equation (2.3), and the truth value of each fuzzy rule is computed by T_3-norm operation as Equation (3.1).

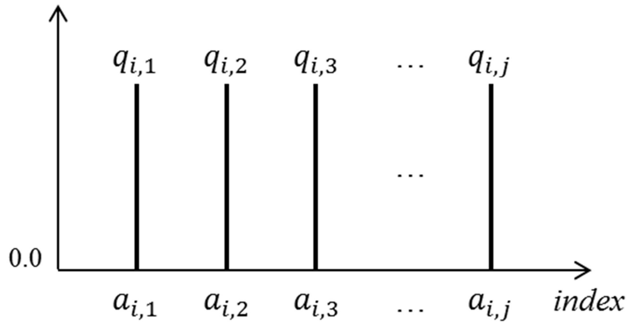
$$v_i(x) = T_3\left(\mu^i_{L_{s_1}}(s_1), \mu^i_{L_{s_2}}(s_2), \dots, \mu^i_{L_{s_n}}(s_n)\right) = \prod_{k=1}^{n} \mu^i_{L_{s_k}}(s_k) \tag{3.1}$$

where,

$\mu_{L^i_{s_n}}$ a membership degree computed by Equation (2.2) for the observed state value s_n of i-th fuzzy rule.

The conclusion part of 0-order TSK model consists of a set of singleton values such as local action values and local q-values. The local singleton q-values are shown in Figure 3.2.2. All of the local q-values are initially zero, and these values are gradually increasing or decreasing during the learning process. Also the local action in fuzzy rule R_i is selected by depending on local q-values. We use

EEP method for selecting the local action. The suggested algorithm of EEP is shown in Figure 3.2.3. A value of ε is set in the beginning of algorithm. In general, the interval of ε is between 0 and 1, and this value is multiplied by β that is greater than 0 and less than or equal to 1 in every episode.



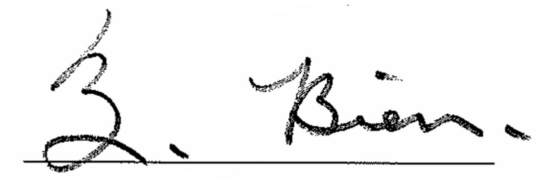


Figure 3.2.2: The local singleton q-values in the conclusion part.

$Generate\ a\ value\ \alpha\ drawn\ from\ standard\ uniform\ distribution$
$\boldsymbol{if}\ \alpha\ is\ greater\ than\ \varepsilon$
$\quad a_{i,k'} = \arg\max_{k \le j} q_{i,k}$
$\quad \boldsymbol{if}\ a_{i,k'}\ is\ vector$
$\quad\quad Choose\ a_{i,j'}\ randomly\ in\ vector\ a_{i,k'}$
$\quad \boldsymbol{else}$
$\quad\quad a_{i,j'} = a_{i,k'}$
$\boldsymbol{else}$
$\quad Choose\ a_{i,j'}\ randomly\ in\ vector\ a_{i,k}, for\ k = 1, \ldots, j$

Figure 3.2.3: The EEP algorithm for selecting the local action.

The global action is computed by truth values and selected local actions as Equation (2.9), and it is considered as decision-making of FQL method. Also, the global action is the input data of ANFIS. The ANFIS method compute the actual output values of the agent based on the global action of FQL. Then, the agent receives the reinforcement signal from an environment. Also, the reinforcement signal affects the system performance. We use the trapezoidal membership shape to infer the reinforcement signal. It is shown in Figure 3.2.4. The reinforcement signal might be positive real number or negative real number. The local q-values of selected local actions in every fuzzy rule are updated by Equation (2.10).

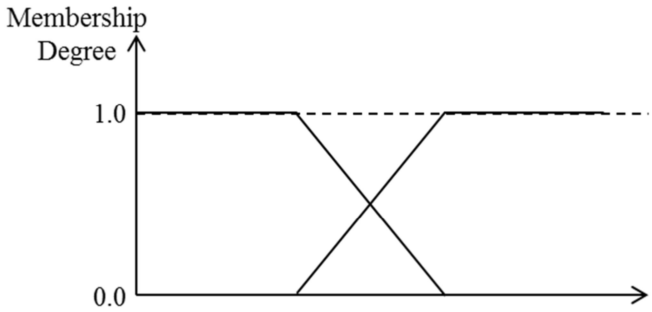


Figure 3.2.4: The trapezoidal membership shape for reinforcement signal.

B. Design of ANFIS

Here, we specify the ANFIS method in this section. The system architecture of ANFIS is following as Figure 2.2.6 to compute the output values. We use the 1-oreder TSK type fuzzy model in ANFIS method. The general form of TSK type fuzzy model is represented as Equation (2.5). The crisp functions of the conclusion part represent 1-order equations regarding input vector as Example 6. Premise parameters and consequent parameters are updated by LSE and gradient descent methods respectively, and the derivative information is used in gradient descent method. So, we use the Bell membership function which is differentiable. It is shown in Figure 3.2.5.

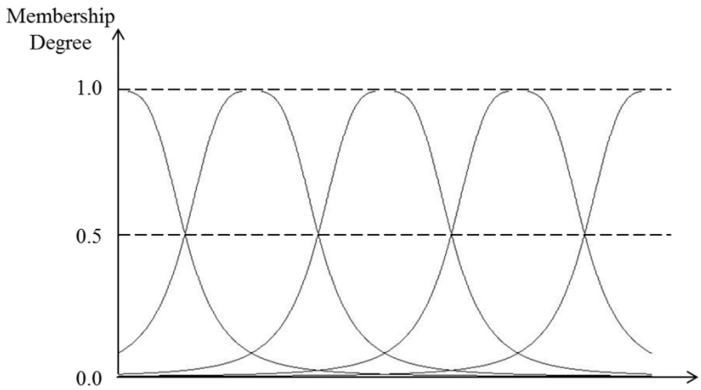


Figure 3.2.5: The Bell membership function shape.

The membership degree of the Bell function is computed as Equation (3.2) regarding the given input value x in layer 1 of the system architecture of ANFIS.

$$\mu_x^i(x) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}} \tag{3.2}$$

where,

a, b, c the parameter set of the Bell membership function.

The Bell function has a set of parameters that is able to change the function shape and location. So, the purpose of gradient descent method is to update the parameter set of Bell function, and these parameters are considered as premise parameters. The variation of membership shapes caused by premise parameter is shown in Figure. 3.2.6. The variance width and slopes is changed by parameters *a* and *b*, and the position of the center of the Bell function depends on the parameter *c*.

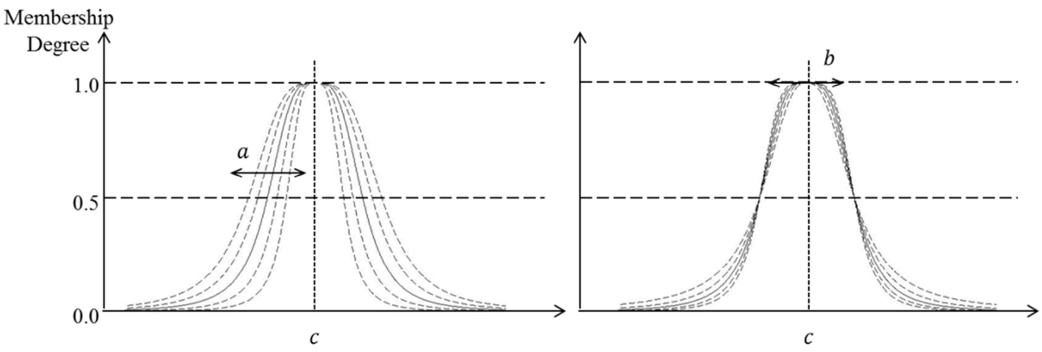


Figure 3.2.6: The change of membership function shape caused by parameters.

The truth value of each fuzzy rule is computed by T_3-norm operation as Equation (3.1) in layer 2. We use the same method with FQL to get the truth value. The normalized truth values are computed in every node in layer 3 as Equation (2.11). Every node in layer 4, the output of i-th node is calculated as Equation (2.12), and finally, the overall outputs are represented in layer 5. In general, the look-up table of FQL is updated by experiencing a state-action pair in every step. So, the FQL is an on-line learning method. Also, the premise and consequent parameters of ANFIS should be updated in on-line regarding a state-action pair. We use the hybrid learning method to update the premise and consequent parameters. The error signal E_p which is a difference between the actual output and the *p*-th desired output is used for updating the premise parameters. Strictly speaking, it is not formal gradient descent method. However it is possible to be approximate when the learning rate is small. The on-line version of sequential LSE method is used to update the consequent parameters. Both parameters are updated after each state-action presentation based on the desired output. Is is estimated based on the observed states in every step.

The sequential LSE method that is Equation (2.16) is modified to on-line version. It is expressed as Equation (3.3).

$$X_{i+1} = X_i + S_{i+1} a_{i+1} \left(b_{i+1}^T - a_{i+1}^T X_i \right) \tag{3.3}$$

$$S_{i+1} = \frac{1}{\lambda}\left(S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{\lambda + a_{i+1}^T S_i a_{i+1}}\right), i = 0, 1, \ldots, P-1$$

where,

λ a forgetting factor that is between 0.9 and 1.

In general, S_i and X_i are initially the identity matrix which has a large positive number and zero, respectively.

The learning process of gradient descent method is shown in Figure 3.2.7.

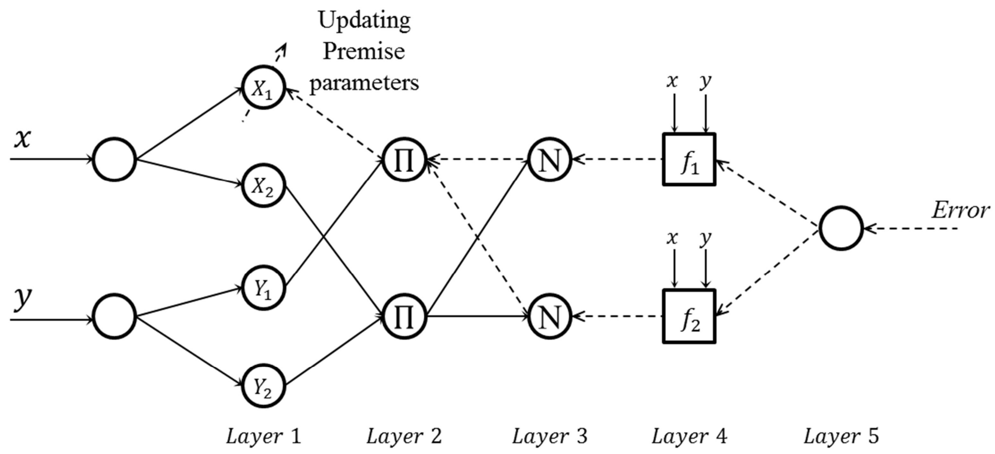


Figure 3.2.7: Updating premise parameters by gradient descent method.

Following paths of the error back propagation, the formula of updating premise parameters is presented as Equation (3.4). The error is back propagated from output layer to first layer. We assume that the number of overall output of system is one.

$$\frac{\partial E_p}{\partial \alpha_i} = \frac{\partial E_p}{\partial O} \cdot \frac{\partial O}{\partial o_i} \cdot \frac{\partial o_i}{\partial w_i} \cdot \frac{\partial w_i}{\partial \mu_{x_n}^i} \cdot \frac{\partial \mu_{x_n}^i}{\partial \alpha_i} \tag{3.4}$$

The partial derivative of each part is derived below:

$$\frac{\partial E_p}{\partial O} = \frac{\partial}{\partial O}(d_i - O)^2 = -2(d_i - O)$$

$$\frac{\partial O}{\partial o_i} = \frac{\partial}{\partial o_i}\sum_{i=1}^{n} o_i = 1$$

$$\frac{\partial o_i}{\partial w_i} = \frac{\partial}{\partial w_i}\overline{w}_i f_i = \frac{f_i - O}{\sum_{i=1}^{n} w_i}$$

$$\frac{\partial w_i}{\partial \mu_{x_n}^i} = \frac{\partial}{\partial \mu_{x_n}^i} \prod_{k=1}^{n} \mu_{x_k}^i(x_k)$$

From Equation (3.2), the partial derivative of each parameter of Bell membership function is derived below:

$$\frac{\partial \mu_{x_n}^i}{\partial a_i} = \frac{2b_i}{a_i} \mu_{x_n}^i (1 - \mu_{x_n}^i)$$

$$\frac{\partial \mu_{x_n}^i}{\partial b_i} = \begin{cases} -2\, ln \left|\frac{x_n - c_i}{a_i}\right| \mu_{x_n}^i \left(1 - \mu_{x_n}^i\right) & if\ x_n \neq c_i \\ 0 & otherwise \end{cases}$$

$$\frac{\partial \mu_{x_n}^i}{\partial c_i} = \begin{cases} \frac{2b_i}{x_n - c_i} \mu_{x_n}^i \left(1 - \mu_{x_n}^i\right) & if\ x_n \neq c_i \\ 0 & otherwise \end{cases}$$

So, Δa_i, Δb_i, and Δc_i are expressed as Equation (3.5)

$$\Delta a_i = -\eta_a \frac{\partial E_p}{\partial a_i}, \qquad \Delta b_i = -\eta_b \frac{\partial E_p}{\partial b_i}, \qquad \Delta c_i = -\eta_c \frac{\partial E_p}{\partial c_i} \tag{3.5}$$

The system flow chart of FQL with ANFIS is shown in Figure 3.2.8.

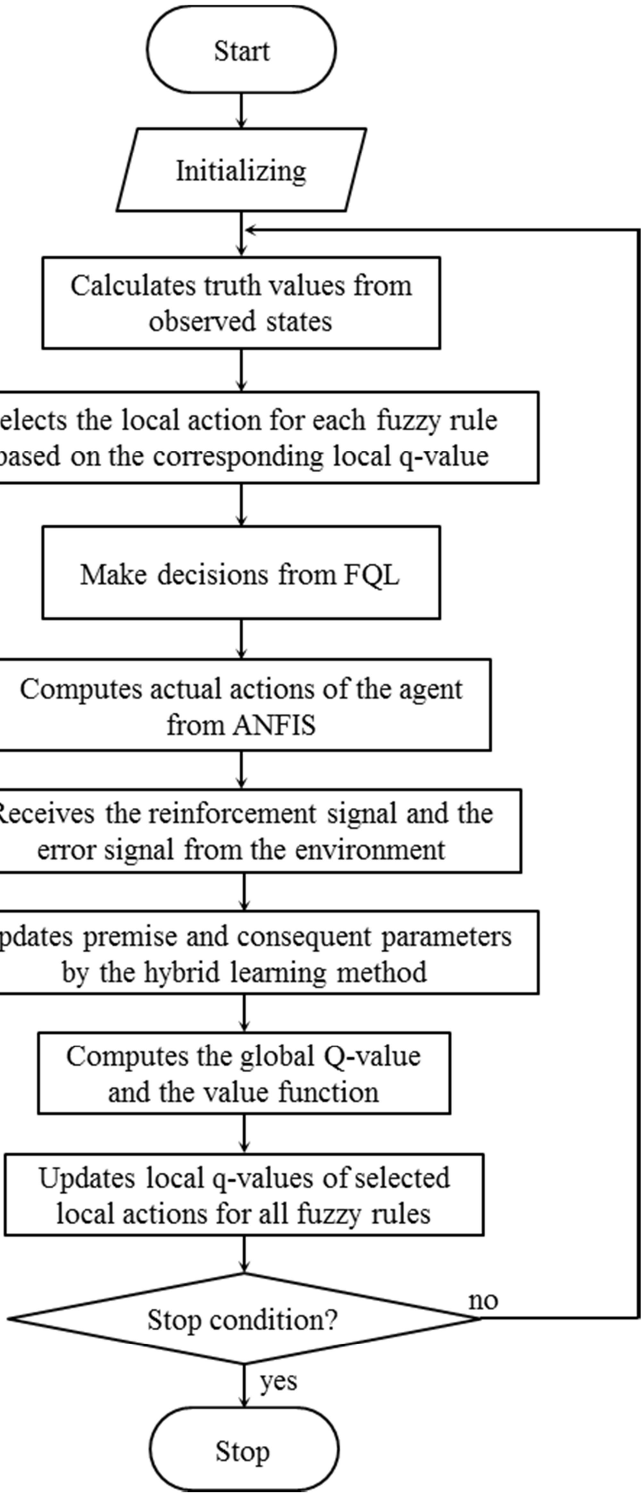


Figure 3.2.8: The system flow chart of FQL with ANFIS.

IV. SIMULATION

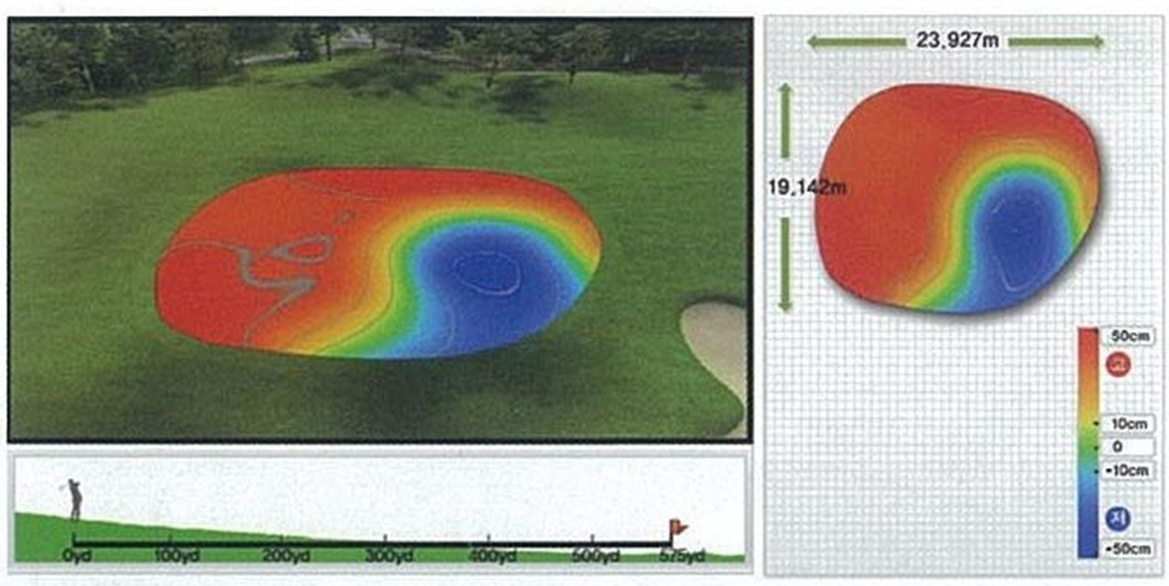


Figure 4.1: The advance information of the putting green in real world.

We implement the simulation of FQL with ANFIS in the putting green environment. There is a hill in the putting green, and it is located between a start position and a goal position. The start position and the goal position are given arbitrarily. The purpose of the agent is to make a hole in one in the putting green environment. We make some constraints in simulation rules.

· Every putting should be conducted in the start position.
· The episode is terminated when the agent is successful to make a hole in one.

The agent conducts the putting in the start position only even if the ball is out of the putting green and fails to make a hole in one. We count the number of putting until the episode is terminated. After several episodes, the counted number of putting regarding all episodes are plotted to confirm the learning efficiency.

A. The definition of states

An example of the putting green in the real world is shown in Figure 4.1. In advance, we can easily know the basic information regarding the putting green, such as the surface of green, the highest point, the lowest point, the area, etc. This information is considered as the partially known information about the environment. Thus, the agent is possible to use some information which is partially known in advance for making decisions. In point of view, we define five observed states, which are based on the partially known information, as below:

x_1: the distance between a start position and a goal position,

x_2: the condition of the grass,

x_3: the inclination of a hill,

x_4: the direction between a start position and a goal position,

x_5: the location of a hill.

B. The putting green environment

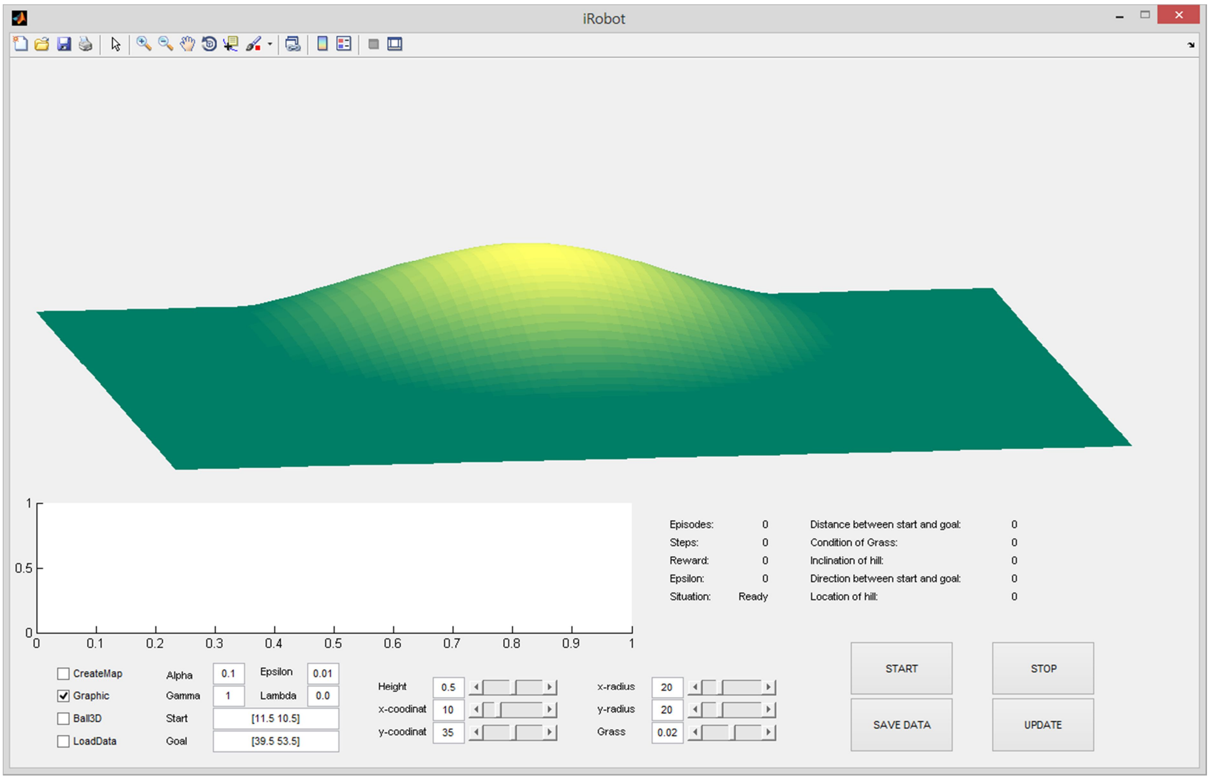


Figure 4.2: The modeled putting green and GUI of FQL with ANFIS.

We model a simple putting green which has a hill. The hill is located between the start position and the goal position arbitrarily. So, the agent should have considered the hill when making decision. The modeled putting green and GUI are shown in Figure 4.2. We use the multivariate normal distribution to make the hill. Its formula is expressed as Equation (4.1).

$$f(x) = h * exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{4.1}$$

where,

x the size of the putting green,

h the height of the hill,

μ the center position of the hill,

Σ the area of the hill.

We constrain the maximum value of the height of the hill is one meter, and set that the width and the length of the putting green are five and seven meters, respectively. In general, humans have inferred the putting direction and strength when they are playing the game of golf in putting green. The agent also should have make decisions such as the putting direction and strength from given states. However, there is low correlation between the putting direction and observed states x_1 and x_2. Similarly, there is low correlation between the putting strength and observed states x_4 and x_5. Thus, we thoroughly decompose a process of the putting direction and a process of the putting strength. On the other hand, the observed state x_3 is correlating with both decisions. So, we have inferred the putting strength and direction based on the observed states x_1, x_2, and x_3, and the observed states x_3, x_4, and x_5, respectively.

C. The fuzzy rules

We define the linguistic words of observed states as below:

x_1: [*Very Short*$(\tilde{3})$ *Short*$(\widetilde{23.3})$ *Normal*$(\widetilde{44.6})$ *Long*$(\widetilde{65.3})$ *Very Long*$(\widetilde{86})$],

x_2: [*Fast*$(\tilde{0})$ *Normal* $(\widetilde{0.5})$ *Slow*$(\tilde{1})$],

x_3: [*Flat*$(\tilde{0})$ *Little High*$(\widetilde{0.5})$ *High*$(\tilde{1})$],

x_4: [*Most Right*$(\tilde{0})$ *Right*$(\widetilde{45})$ *Middle*$(\widetilde{90})$ *Left*$(\widetilde{135})$ *Most Left*$(\widetilde{180})$],

x_5: [*Left*$(\widetilde{-1})$ *Middle*$(\tilde{0})$ *Right*$(\tilde{1})$].

As already mentioned, we use the triangular membership function that is shown in Figure 3.2.1. So, there are forty-five fuzzy rules of each decomposed part. The fuzzy rules of FQL which consist of precondition and conclusion parts are expressed as below and Table 2 when the number of singleton of local actions and local q-values is three.

R_{s_1}: **IF** x_1 is $Very\ Short$ and x_2 is $Fast$ and x_3 is $Flat$

THEN $a_{s_{1,1}}$ with $q_{s_{1,1}}$ or $a_{s_{1,2}}$ with $q_{s_{1,2}}$ or $a_{s_{1,3}}$ with $q_{s_{1,3}}$

R_{s_2}: **IF** x_1 is $Very\ Short$ and x_2 is $Fast$ and x_3 is $Little\ High$

THEN $a_{s_{2,1}}$ with $q_{s_{2,1}}$ or $a_{s_{2,2}}$ with $q_{s_{2,2}}$ or $a_{s_{2,3}}$ with $q_{s_{2,3}}$

$\vdots$

$R_{s_{45}}$: **IF** x_1 is $Very\ Long$ and x_2 is $Slow$ and x_3 is $High$

THEN $a_{s_{45,1}}$ with $q_{s_{45,1}}$ or $a_{s_{45,2}}$ with $q_{s_{45,2}}$ or $a_{s_{45,3}}$ with $q_{s_{45,3}}$

R_{d_1}: **IF** x_3 is $Flat$ and x_4 is $Most\ Right$ and x_5 is $Left$

THEN $a_{d_{1,1}}$ with $q_{d_{1,1}}$ or $a_{d_{1,2}}$ with $q_{d_{1,2}}$ or $a_{d_{1,3}}$ with $q_{d_{1,3}}$

R_{d_1}: **IF** x_3 is $Flat$ and x_4 is $Most\ Right$ and x_5 is $Middle$

THEN $a_{d_{2,1}}$ with $q_{d_{2,1}}$ or $a_{d_{2,2}}$ with $q_{d_{2,2}}$ or $a_{d_{2,3}}$ with $q_{d_{2,3}}$

$\vdots$

$R_{d_{45}}$: **IF** x_3 is $High$ and x_4 is $Most\ Left$ and x_5 is $Right$

THEN $a_{d_{45,1}}$ with $q_{d_{45,1}}$ or $a_{d_{45,2}}$ with $q_{d_{45,2}}$ or $a_{d_{45,3}}$ with $q_{d_{45,3}}$

The condition of grass / **Inclination** / **Distance**	***Fast***			***Normal***			***Slow***			
	Flat	***Little High***	***High***	***Flat***	***Little High***	***High***	***Flat***	***Little High***	***High***	
Very Short	$q_{s_{1,j}}$	$q_{s_{2,j}}$	$q_{s_{3,j}}$	…	…	…	$q_{s_{7,j}}$	$q_{s_{8,j}}$	$q_{s_{9,j}}$	The putting strength
Short	$q_{s_{10,j}}$	…	…	⋱			…	…	$q_{s_{18,j}}$	
Normal	⋮				⋱				⋮	
Long	$q_{s_{28,j}}$	…	…			⋱	…	…	$q_{s_{36,j}}$	
Very Long	$q_{s_{37,j}}$	$q_{s_{38,j}}$	$q_{s_{39,j}}$	…	…	…	$q_{s_{43,j}}$	$q_{s_{44,j}}$	$q_{s_{45,j}}$	
Location of hill / **Inclination** / **Direction**	***Left***			***Middle***			***Right***			
	Flat	***Little High***	***High***	***Flat***	***Little High***	***High***	***Flat***	***Little High***	***High***	The putting direction
Most Right	$q_{d_{1,k}}$	$q_{d_{2,k}}$	$q_{d_{3,k}}$	…	…	…	$q_{d_{7,k}}$	$q_{d_{8,k}}$	$q_{d_{9,k}}$	

Right	$q_{d_{10,k}}$	$\cdots$	$\cdots$	$\ddots$			$\cdots$	$\cdots$	$q_{d_{18,k}}$	
Middle	$\vdots$				$\ddots$				$\vdots$	
Left	$q_{d_{28,k}}$	$\cdots$	$\cdots$			$\ddots$	$\cdots$	$\cdots$	$q_{d_{36,k}}$	
Most Left	$q_{d_{37,k}}$	$q_{d_{38,k}}$	$q_{d_{39,k}}$	$\cdots$	$\cdots$	$\cdots$	$q_{d_{43,k}}$	$q_{d_{44,k}}$	$q_{d_{45,k}}$	

Table 4: The fuzzy rules of FQL.

where, j=1, 2, 3 and k=1, 2, 3.

As the fuzzy rules, if there are three singleton values of strength part and three singleton values of direction part, the agent

From the fuzzy rules of FQL, we are able to know that the agent make decisions, the putting strength and the putting direction. Thus, the ANFIS should compute the actual direction and the actual velocity of the motors. We also decompose the ANFIS as velocity part and direction part. It is shown in Figure 4.3 when we use three linguistic words in both parts.

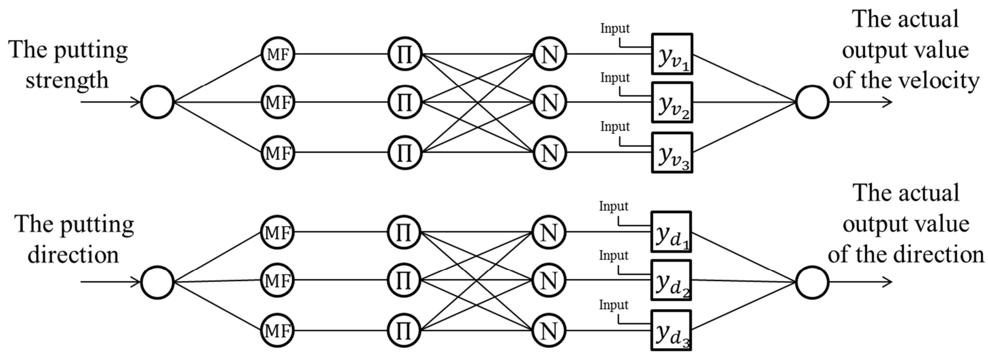


Figure 4.3: The structures of ANFIS regarding the actual outputs of velocity and direction.

D. Rewards

The reinforcement signals are represented as three cases as following:

· Case 1: The ball is in the hole.

· Case 2: The ball is out of the putting green.

· Case 3: The ball is in the putting green.

The case 1 is that the agent is successful to make a hole in one. In this case, the agent receives the reinforcement signal that is a positive number as 1. On the other hand, the agent receives the reinforcement signal that is a negative number as -1 when the ball is out of the putting green. Finally, in case3, the agent receives the reinforcement signal which depends on differences between the ball position caused by actual action and the goal position. We use fuzzy variables to get the reinforcement signals regarding the putting strength and direction. The fuzzy variables are shown in Figure 4.4. The differences such as the distance and the direction are mapped from -1 to 1. The mapped values become to input values for inferring the reinforcement signals. This signal is denoted as the multiplication of membership degree and -1. So, the interval of the reinforcement signal of case 3 is between 0 and -1.

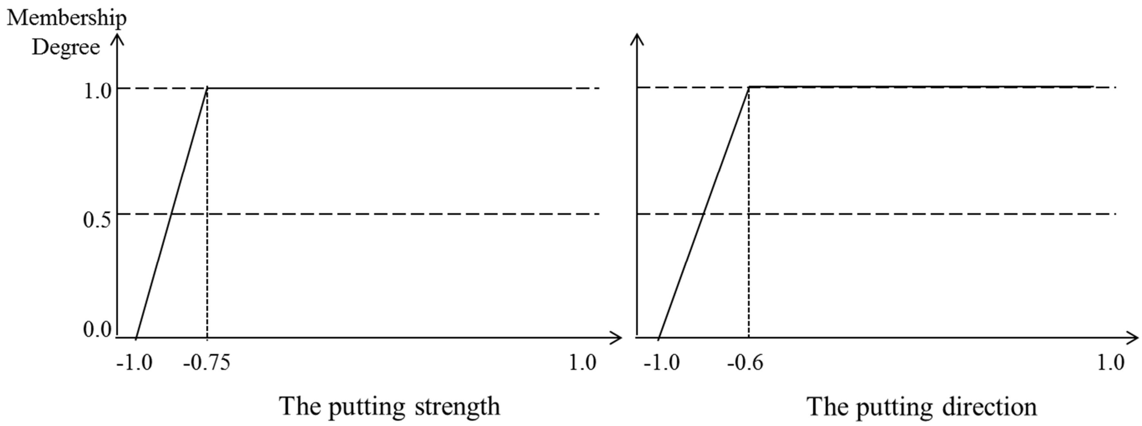


Figure 4.4: The fuzzy variables of the putting strength and direction.

E. The putting simulation

We implement the simulation of FQL with ANFIS method to be applied the hole in one system in the putting green. The putting green is built randomly. The maximum number of episodes is restricted as twenty, and the maximum number of the putting in each episode is a thousand. The membership degrees of observed states are denoted as Table 3.

The distance between a start position and a goal position	**The condition of the grass**	**The inclination of the hill**	**The direction between a start position and a goal position**	**The location of the hill**
Normal(0.67) Long(0.33)	Normal(1.0)	Flat(0.61) Little High(0.39)	Left(0.73) Middle(0.27)	Left(0.61) Middle(0.39)

Table 3: The membership degrees of observed states.

Firstly, we implement the ordinary FQL method. The initial parameters of Q-Learning are set as below:

· the learning rate: 0.1,

· the discount rate: 1.0,

· the EEP factor (ε): 0.01,

· the number of the putting strength singleton values: 30,

・ the number of the putting direction singleton values: 30.

These values are determined experimentally. We implemented the simulation several times for ordinary Q-Learning and the best 3 cases are shown in Figure 4.5 and Table 4.

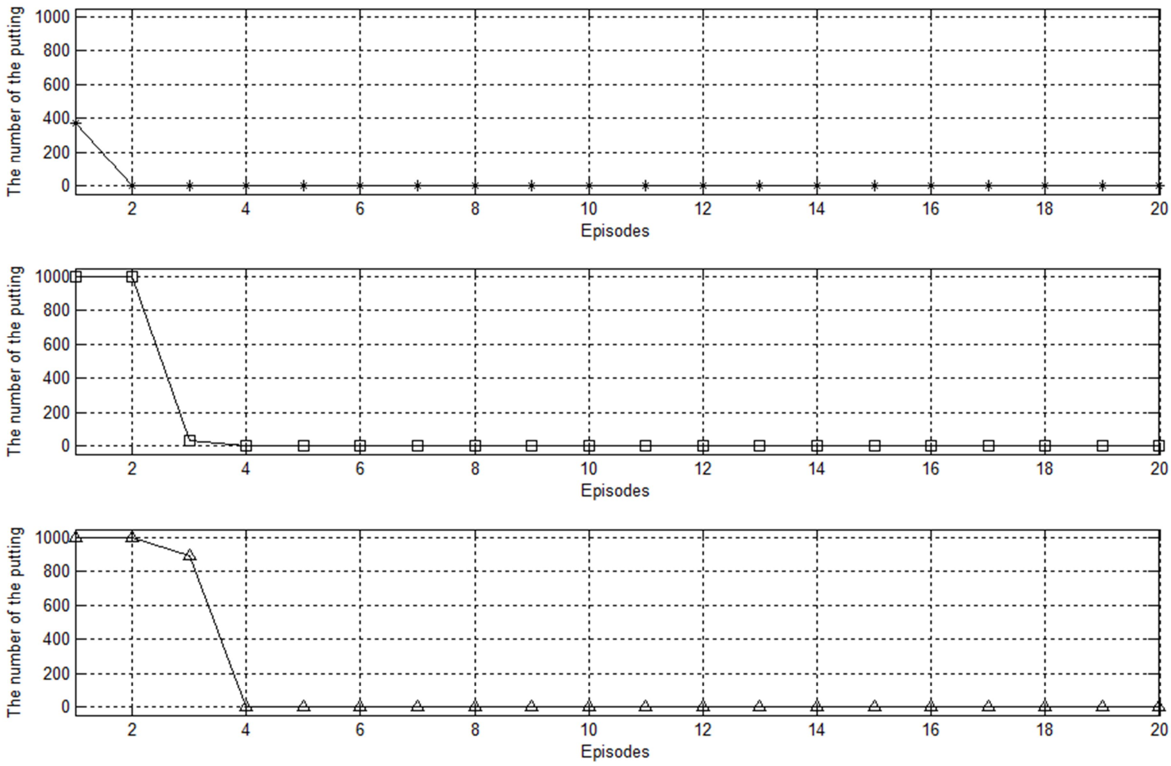


Figure 4.5: The best three cases of ordinary Q-Learning.

	The total number of the putting before making a hole in one	The first hole in one episode
Case 1	372	2
Case 2	2032	4
Case 3	2895	4

Table 4: The best three cases of ordinary Q-Learning.

It takes a lot of time to find the optimal behavior for ordinary Q-Learning even if it is fail to find an actual direction and an actual velocity when the number of singleton values is small. Now, we conduct the FQL with ANFIS method as the same conditions. The initial parameters of the proposed method are set as below:

・ the learning rate of FQL: 0.1,

・ the discount rate: 1.0,

· the EEP factor (ε): 0.01,

· the learning rate of the gradient descent for strength part: 0.1

· the learning rate of the gradient descent for direction part: 0.5

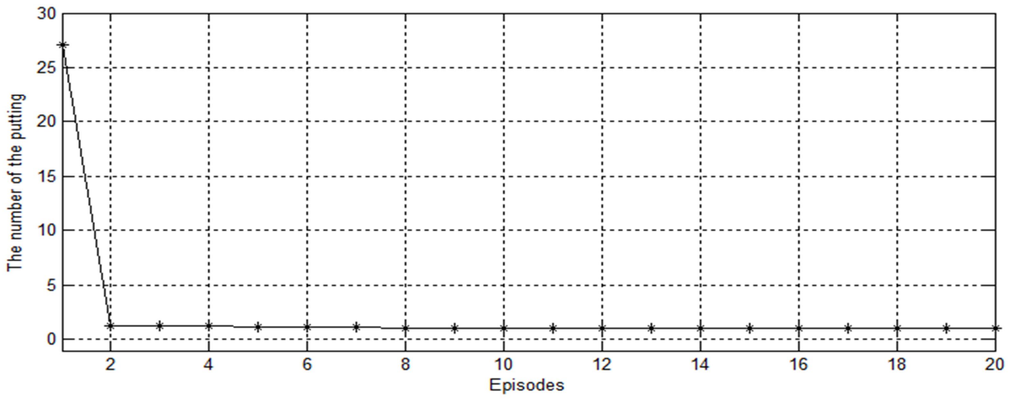
· the number of the putting strength singleton values: 2,

· the number of the putting direction singleton values: 2,

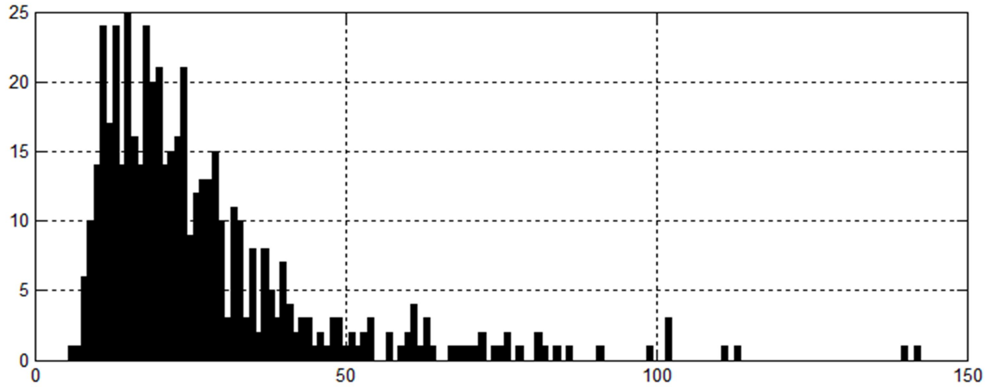
· the number of the linguistic words of strength part of ANFIS: 3,

· the number of the linguistic words of direction part of ANFIS: 3.

These learning parameters are set experimentally. We iterate five-hundred times the simulation and the result of the average value is shown in Figure 4.6 and Table5.



(a)



(b)

Figure 4.6: (a) The average number of the putting when the number of decisions is four. (b) The histogram of the first episode.

	The number of iteration	The minimum putting	The maximum putting	The average of putting in first episode
2 x 2 decisions and 3 linguistic words	500	6	142	27.11

Table 5: The simulation result of 2 x 2 decisions FQL with 3 linguistic words of ANFIS.

Most of the iterations, the agent made the hole in one after the first episode. Next, we only increase the number of the linguistic words of both parts in ANFIS from three to five. It is shown in Table 6.

	The number of iteration	The minimum putting	The maximum putting	The average of putting in first episode
2 x 2 decisions and 5 linguistic words	500	6	165	26.56

Table 6: The simulation result of 2 x 2 decisions FQL with 5 linguistic words of ANFIS.

Most of the iterations, the agent made the hole in one after the first episode. Next, we only increase the number of the linguistic words of both parts in ANFIS from three to five. It is shown in Table 6. We compare the results in Table 5 and Table6. These results are similar. It means that there are no large differences between the number of three and five linguistic words.

Now, we implement the 1 x 1 decision. It is considered as a simple ANFIS since the decision-making values are always the maximum values. The result is shown in Figure 4.7 and Figure 4.8.

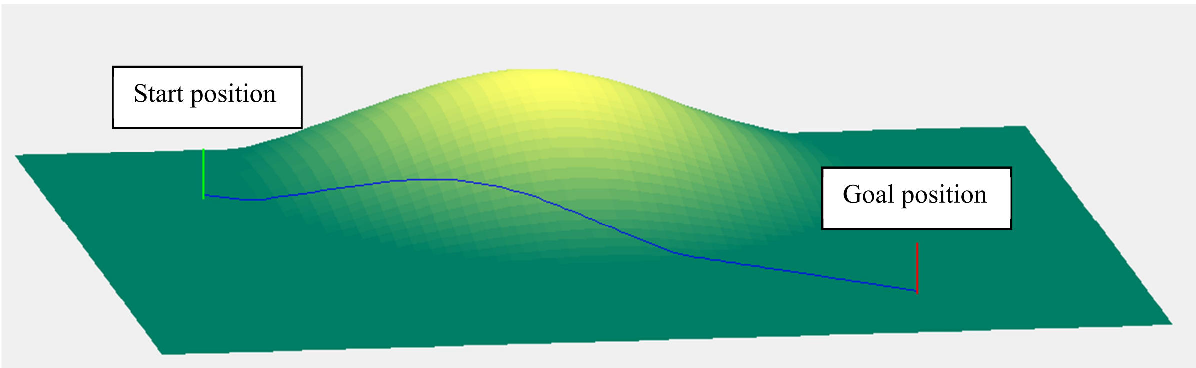


Figure 4.7: The trajectory of the ball when making a hole in one.

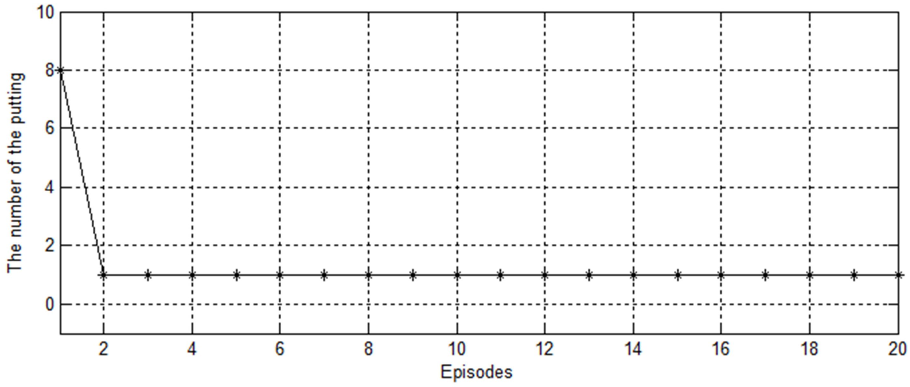
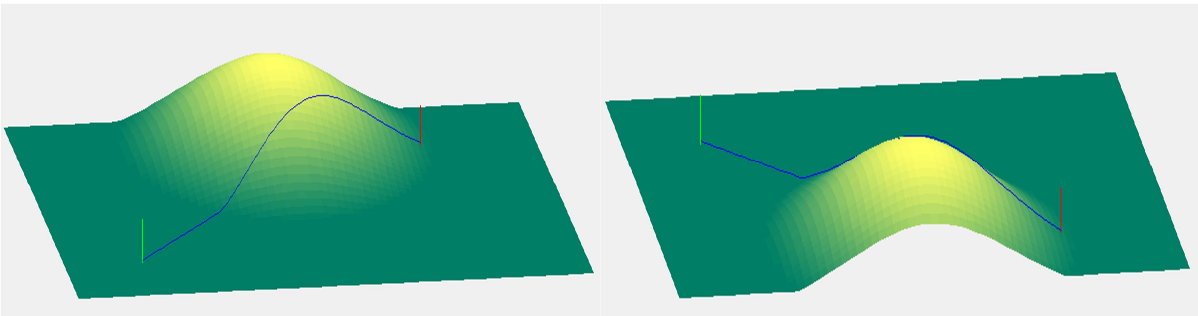


Figure 4.8: The number of the putting when the decision is 1 x 1.

The learning structure of FQL with ANFIS is that the actual output is calculated by adapting premise and consequent parameters based on decision-making. So, we are able to expect to improve the learning time by using the data which is already learned in similar environment beforehand. We set that the number of decision is four, and the agent learned the two kinds of situations in advance as Figure 4.9.



(a) (b)

Figure 4.9: (a) The first situation. (b) The second situation.

We use the putting green which has two hills, and the shape of the putting green is curve. The behavior of the agent is shown in Figure 4.10. It is hard to make a hole in one since the shape of the putting ground. The agent makes a goal after 18th putting when using the learned data as shown Figure 4.9. On the other hand, the agent makes a goal after 872th putting, which is the best result, when the learned data is not used.

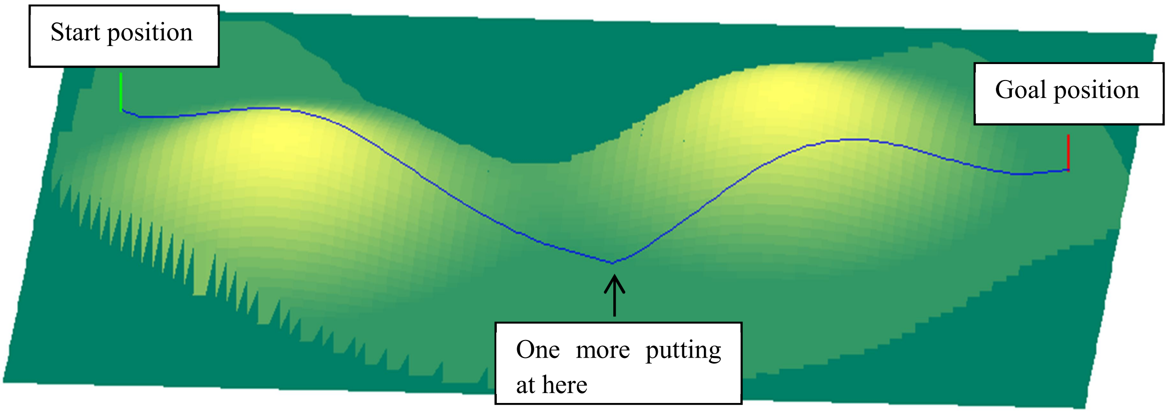


Figure 4.10: The human behavior learning based on using the learned data in the similar environment.

V. CONCLUSION

In this thesis, we introduced the integrated fuzzy logic system FQL with ANFIS to learn the human behaviors. Humans recognize their situations and infer the behaviors for achieving an aim. This process of human reasoning is described as fuzzy rules and the agent uses this information as *a priori knowledge* to find an optimal behavior. We presented FQL method examples such as the mountain car problem and the navigation problem. Through these examples, we mentioned the difficulties to expect the efficiency of learning, and suggested the FQL with ANFIS. In the FQL method, the agent makes decision based on fuzzy rules. The ANFIS is used to the adaptive controller for computing the actual output values of the agent based on decision-making. Moreover this method is simulated to be applied to the hole in one system in the putting green to mimic the human behavior and to find the precise actual action values.

The integrated fuzzy logic system which is the FQL with the ANFIS is useful for solving the problem that is required for strategies and precise actions. There are a lot of ways to combine the FQL and the ANFIS, and one of them was mentioned in this thesis. We performed the simulation of the proposed method by applying the game of the putting, and compared the performance of the FQL and the proposed method. However, the proposed method is not perfect. For example, the performance of the proposed method is highly sensitive regarding the number of decisions. We are able to expect the better performance for finding an optimal behavior and mimicking the process of human reasoning by studying the other combination ways.

REFERENCE

1. Brumitt, B and Meyers, B and Krumm, J and Kern, A and Shafer, S 2000, “Easy Living: Technologies for Intelligent Environment”, *International Symposium on Handheld and Ubiquitous Computing*, Bristol, UK, pp. 12-29, September.

2. Bien, ZZ and Lee, HE and Do, JH and Kim, YH and Park, KH and Yang, SE 2008, “Intelligent Interaction for Human-friendly Service Robot in Smart House Environment”, International Journal of Computational Intelligence Systems, vol. 1, no. 1, pp. 77-93, January.

3. Bien, ZZ 2011, “Intelligent System”, Lecture Material, Electrical Engineering, Ulsan National Institute of Science and Technology.

4. Efendigil, T and Onut, S and Kahraman, C 2009, “A Decision Support System for Demand Forecasting with Artificial Neural Networks and Neuro-Fuzzy Models: A Comparative Analysis”, Expert Systems with Applications, vol. 36, pp. 6697-6707.

5. Glorennec, PY and Jouffe L 1997, “Fuzzy Q-Learning”, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 659-662.

6. House_n, http://architecture.mit.edu/house_n.

7. Jang, J-SR 1991, “Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm”, *AAAI’91 Proceedings of the ninth National conference on Artificial Intelligence*, vol. 2, pp. 762-767.

8. Jang, J-SR 1993, “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, *IEEE Transactions on Systems*, Man and Cybernetics, vol. 23, no. 3.

9. Jang, J-SR and Sun, C-T and Mizutani, E 1991, “Neuro-Fuzzy and Soft Computing”, Prentice Hall.

10. Jouffe, L 1998, “Fuzzy Inference System Learning by Reinforcement Methods”, *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 28, no. 3, August.

11. Kaelbling, LP and Litman, ML and Moore, AW 1996, “Reinforcement Learning: A Survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285.

12. Kohonen, T 1982, “Self-Organized Formation of Topologically Correct Feature Maps”, *Biological*

Cybernetics, vol. 43, pp. 59-69.

13. Lee, H-E and Kim, Y and Park, K-H and Bien, ZZ 2006, “Development of a Steward Robot for Human-friendly Interaction”, *Proceeding of the IEEE International Conference on Control Applications*, Munich, Germany, pp. 551-556, October.

14. Lee, J-J and Seo, K-H and Oh, C and Bien, ZZ 2007, “Development of a future Intelligent Sweet Home for the disabled”, *International Symposium on Artificial Life and Robotics*, pp. 23-25, January.

15. Lee, SW 2008, “Parameter Optimization for Kernel-based Pattern Classification and Agglomerative Clustering”, Ph.D. dissertation, Korea Advanced Institute of Science and Technology, Daejeon, Korea.

16. Moody, J and Saffell, M 2001, “Learning to Trade via Direct Reinforcement”, *IEEE Transactions on Neural Networks,* vol. 12, no. 4, pp. 875-889, July.

17. Mozer, MC 1998, “The neural network house: An environment that adapts to its inhabitants”, *The American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, pp. 110- 114.

18. Park, K-H and Lee, H-E and Kim, Y and Bien, ZZ 2007, “A Steward Robot for Human-friendly Human-Machine Interaction in a Smart House Environment”, *IEEE Transactions on Automation Science and Engineering*, vol. 5, Issue 1, pp. 21-25, January.

19. Rummery, GA and Niranjan, M 1994, “On-Line Q-Learning Using Connectionist Systems”, *Technical Report CUED/F-INFENG/TR 166*, Cambridge University Engineering Department, September.

20. Smith, AJ 2002, “Application of the Self-Organizing Map to Reinforcement Learning”, Neural Networks, vol. 15, Issues 8-9, pp. 1107-1124.

21. Sutton, RS 1988, “Learning to Predict by the Methods of Temporal Differences”, *Machine Learning*, vol. 3, pp. 9-44.

22. Sutton, RS 1996, “Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding”, *Advances in Neural Information Processing Systems*, vol. 8, pp. 1038-1044, MIT Press.

23. Sutton, RS and Barto, AG 1998, “Reinforcement Learning: An Introduction”, MIT Press, Cambridge, MA.

24. Watkins, CJCH 1989, “Learning from Delayed Reward” Ph.D. dissertation, Cambridge University, Cambridge, U.K.

25. Wender, S and Watson, I 2008, “Using Reinforcement Learning for City Site Selection in the Turn-Based Strategy Game Civilization IV”, *CIG’08. IEEE Symposium on Computational Intelligence and Games*, pp. 372-377, December.

26. WHO and World Bank 2011, “World Report on Disability”, Report.

27. Yang, Q and Vance, JB and Jagannathan, S 2008, “Control of Nonaffine Nonlinear Discrete-Time Systems Using Reinforcement-Learning-Based Linearly Parameterized Neural Networks”, *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics,* vol. 38, no. 4, pp. 994-1001, August.

ACKNOWLEDGEMENTS

First of all, I would really like to express my sincere gratitude to my advisor Prof. Z. Zenn Bien who is a chair professor of Electrical Engineering program of UNIST for giving me the opportunity to study the intelligent control system and the continuous support of research for my Master Course. It is very variable experience for me to be under his competent guidance. Also, his wide knowledge regarding intelligent control filed and diverse experience have been great value for me. Thanks to give a lot of advice to me for two years, I was able to complete my study and thesis successfully. He always emphasizes the importance of English and Question for me and I would appreciate his guidance because I could improve my English ability and learn how to think.

Besides, I really wish to express my sincere appreciate to the thesis committee members which are Prof. Gil-Jin Jang of Machine Intelligence lab and Prof. Jae-Young Sim of Visual Information Processing lab. Their insightful comments and evaluation were very helpful for improvement of my career.

I would like to thank my roommate Kangyeoul Lee who is a member of MI lab for giving a pleasant time to me. I couldn't forget the time of taken late-night meals in every night. Thanks to him, I was able to endure the hard time during my Master Course. Also I would like to thank my friends Taehui Yoon who is a member of VIP lab, Yeonghun Jeong, Seongseok Seo, Toan Xuan Mai who are members of Co-design and Parallel Processing lab, and Junyoung Min who is a member of MI lab. Thanks to them, I could enjoy my unist life.

Moreover, I would like to thank my family Sangib Shin, Sunim Choi who are my parents and Hwajin Shin who is my older sister for giving support and encouragement to me during my Master Course. Also they were always cheering up for me. Finally, I would really thank to my girlfriend Youngka Lee. She was always supporting and cheering for me during my Master Course, so I was able to finish it safely.